

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

TESE DE DOUTORADO

**PROJETO E CONSTRUÇÃO DE UM CONVERSOR ANALÓGICO/DIGITAL  
RÁPIDO BIPOLAR TIPO DUPLO *FOLDING* COM NOVAS TÉCNICAS DE  
INTERPOLAÇÃO E CORREÇÃO DE ERRO**

EVANDRO MAZINA MARTINS

Este exemplar corresponde a redação final da tese  
defendida por EVANDRO MAZINA  
MARTINS e aprovada pela Comissão  
julgada em 25 / 11 / 1999.  
  
Orientador

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica.

Banca Examinadora:

Prof. Dr. Elnatan Chagas Ferreira (Orientador)

Prof. Dr. Nobuo Oki - UNESP/Ilha Solteira

Prof. Dr. Laércio Caldeira - EFEI / Itajuba

Prof. Dr. José Antônio Siqueira Dias - FEEC/UNICAMP

Prof. Dr. Wilmar Bueno de Moraes – Credenciado/FEEC/UNICAMP

CAMPINAS  
25 de Novembro de 1999



|              |                                     |
|--------------|-------------------------------------|
| UNIDADE      | Be                                  |
| N.º CHAMADA: | 11/UNICAMP                          |
|              | M366p                               |
| V.           | Ex.                                 |
| TOMBO BC/    | 40534                               |
| PROC.        | 278/00                              |
| C            | <input type="checkbox"/>            |
| D            | <input checked="" type="checkbox"/> |
| PREÇO        | 311,00                              |
| DATA         | 16/03/00                            |
| N.º CPD      |                                     |

CM-00135087-9

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M366p Martins, Evandro Mazina  
Projeto e construção de um conversor analógico/digital rápido bipolar tipo duplo *folding* com novas técnicas de interpolação e correção de erro / Evandro Mazina Martins.--Campinas, SP: [s.n.], 1999.

Orientador: Elnatan Chagas Ferreira.  
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Conversores analógicos-digitais. 2. Interpolação. 3. Bolhas (Física). 4. Código de controle de erros (teoria da informação). I. Ferreira, Elnatan Chagas. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

## RESUMO

Na arquitetura duplo “folding” com interpolação, os bits mais significativos são determinados pela quantização do sinal de entrada usando um circuito “folding” e os bits menos significativos são obtidos pela técnica de interpolação. A maioria das soluções empregadas para implementar uma técnica de interpolação utiliza a interpolação resistiva ou uma técnica de interpolação por divisão de corrente (com transistores NMOS). Estas técnicas de interpolação têm alguns aspectos indesejáveis. Alternativamente, este trabalho propõe fazer a interpolação no circuito de “folding encoder” e nos “latches” mestre do conversor A/D. Os resultados mostraram que a nova técnica de interpolação permite construir conversores A/D de 8 bits, porém é necessário cuidados especiais na determinação das áreas dos transistores que fazem a interpolação dupla.

Nos conversores A/D tipo duplo “folding”, um conjunto de alguns “latches” mestre-escravo transforma a informação analógica interpolada em um código circular. Um erro de decisão em um “latch” mestre-escravo pode causar erro no código circular (denominado erro de bolha). Técnicas de correção de erro detectam e corrigem os erros de bolha, melhorando a razão de erro do conversor A/D. Este trabalho também propõe e descreve um novo método para a técnica digital de correção de erro que detecta e corrige erros de bolha durante a detecção da transição de zero para um do código circular.

Além disso, este trabalho propõe uma nova topologia para o conversor A/D que permite diminuir a complexidade do circuito e o consumo de potência, com a conseqüente redução da área do “chip”.

Palavras-chave: Bolha, Código Circular, Conversor Analógico/Digital, Correção de Erro, “Folding”; Interpolação.

---

## ABSTRACT

In a double folding architecture with interpolation, the most significant bits are determined by the quantization of input signal using a folding circuit and the least significant bits are obtained by interpolation technique. Most of the solutions employed to implement an interpolation technique use a resistive interpolation or a current division interpolation technique (with NMOS transistor). These interpolation techniques have some undesirable features. Alternatively, this work proposes to make the interpolation in the folding encoder circuit and in the master latches of the A/D converter. The results showed that the new interpolation technique allows to build A/D converters of 8 bits, even so it is necessary special care in the determination of the areas of the transistors that make the double interpolation.

In double folding A/D converters, a set of some master-slave latches transforms the interpolated analog information into circular code. A decision error in a master-slave latch may cause error in the circular code (the so-called bubble error). Error correction techniques detect and correct bubble errors improving the error rate of the A/D converter. This work also proposes and describes a new method for digital error correction technique that detects and corrects bubble errors during the transition detection from zero to one for circular code.

Furthermore, this work proposes a new topology for the A/D converter that allows the decrease of circuit complexity and of the potency consumption, with the consequent reduction of the area of the " chip "

Keywords: Bubble, Circular Code, Analog/Digital Converter, Error Correction, Folding, Interpolation.

*Aos meus pais José Leandro e Maria Terezinha e a minha esposa  
Miriam, pelo constante incentivo.*

## **AGRADECIMENTOS**

- A Deus, por mais está graça;
- Aos meus pais e a minha esposa, de quem sempre recebi o apoio e encontrei a motivação para não desistir;
- Ao Prof. Dr. Elnatan Chagas Ferreira pela orientação, paciência e incentivo;

A realização deste trabalho foi possível em função da colaboração direta ou indireta de muitas pessoas. Manifesto minha gratidão a todas elas e de forma especial:

- Aos Profs. Drs. Alberto Martins Jorge e José Antônio Siqueira Dias pelas disciplinas cursadas;
- À Ademilde Félix pela paciência, ajuda e colaboração na revisão no texto, no inglês e na secretaria do DEMIC;
- Ao Paulo Henrique Machado pela colaboração na área de informática e apoio no laboratório;
- À Regina Maria Aparecida Gazzola Floriano pelas placas de circuito impresso;
- Ao Fernando Ferreira de Melo pela ajuda com o LabView;
- Ao Prof. Amâncio Rodrigues da Silva Júnior do Departamento de Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul pelo constante apoio;
- A todos os professores do Departamento de Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul pela confiança em mim depositada;
- À CAPES, através do PICD, pelo apoio financeiro;
- Às demais pessoas que direta ou indiretamente contribuíram para a realização deste trabalho.

*"Se os leves dedos do rádio conseguem tirar da noite uma melodia e lança-la por sobre as montanhas e o mar; se as notas do violino, brancas como pétalas, voam, atravessando o deserto e o ruído da cidade; se se captam canções como rosas rubras no rarefeito éter azul... por que então os mortais ainda duvidam que Deus ouça as nossas orações ?"*

Marvin Drake

# SUMÁRIO

|                                                                   |      |
|-------------------------------------------------------------------|------|
| RESUMO E ABSTRACT .....                                           | II   |
| LISTA DE FIGURAS .....                                            | IX   |
| LISTA DE TABELAS.....                                             | XII  |
| LISTA DE ABREVIATURAS .....                                       | XIII |
| INTRODUÇÃO .....                                                  | 1    |
| CAPÍTULO 1 – A TÉCNICA DUPLO “FOLDING” COM INTERPOLAÇÃO .....     | 7    |
| 1.1 – DESCRIÇÃO DA TÉCNICA.....                                   | 7    |
| 1.2 – AMPLIFICADOR “FOLDING” .....                                | 8    |
| 1.3 – “FOLDING ENCODER” .....                                     | 12   |
| 1.4 – INTERPOLAÇÃO .....                                          | 14   |
| 1.5 – “LATCHES” MESTRE-ESCRAVO .....                              | 17   |
| 1.6 – ERRO DE CÓDIGO .....                                        | 23   |
| 1.7 – CIRCUITO CODIFICADOR .....                                  | 26   |
| 1.8 – SINCRONISMO DE BITS .....                                   | 28   |
| 1.9 – PORTAS DE SAÍDA .....                                       | 31   |
| CAPÍTULO 2 - INTERPOLAÇÃO .....                                   | 33   |
| 2.1 – CARACTERÍSTICAS GERAIS .....                                | 33   |
| 2.2 – INTERPOLAÇÃO RESISTIVA.....                                 | 34   |
| 2.3 – INTERPOLAÇÃO EM MODO DE CORRENTE .....                      | 39   |
| 2.4 – INTERPOLAÇÃO PROPOSTA INICIAL.....                          | 42   |
| 2.4.1 – INTERPOLAÇÃO SIMPLES .....                                | 43   |
| 2.4.2 – INTERPOLAÇÃO DUPLA.....                                   | 44   |
| 2.4.3 – LIMITAÇÕES .....                                          | 46   |
| 2.4.4 – CONCLUSÃO .....                                           | 49   |
| 2.5 – INTERPOLAÇÃO PROPOSTA FINAL .....                           | 49   |
| 2.5.1 – INTERPOLAÇÃO SIMPLES .....                                | 50   |
| 2.5.2 – INTERPOLAÇÃO DUPLA.....                                   | 55   |
| 2.5.3 – LIMITAÇÕES .....                                          | 58   |
| 2.5.4 – CONCLUSÃO .....                                           | 61   |
| CAPÍTULO 3 – CONVERSOR ANALÓGICO DIGITAL DE 8 BITS - PROJETO..... | 62   |
| 3.1 – CARACTERÍSTICAS GERAIS .....                                | 62   |
| 3.2 – AMPLIFICADORES “FOLDING” .....                              | 62   |
| 3.3 – “FOLDING ENCODER” .....                                     | 65   |
| 3.4 – FONTE DE POLARIZAÇÃO .....                                  | 69   |
| 3.5 – CIRCUITO DE “CLOCK” .....                                   | 71   |
| 3.6 – “LATCH” MESTRE.....                                         | 71   |
| 3.6.1 – “LATCH” MESTRE SIMPLES .....                              | 71   |
| 3.6.2 – “LATCH” MESTRE DUPLO .....                                | 73   |
| 3.7 – PORTAS OU-EXCLUSIVO COM “LATCH” .....                       | 76   |
| 3.8 – CODIFICAÇÃO COM CORREÇÃO DE ERRO .....                      | 82   |
| 3.8.1 – CODIFICAÇÃO .....                                         | 82   |
| 3.8.2 – CORREÇÃO DE ERRO DE BOLHA .....                           | 83   |
| 3.8.3 – RESULTADOS E LIMITAÇÕES .....                             | 85   |
| 3.8.4 – SEGUNDA CORREÇÃO DE ERRO .....                            | 87   |

|                                                             |            |
|-------------------------------------------------------------|------------|
| 3.8.5 – ANÁLISE DA CORREÇÃO DE ERRO .....                   | 88         |
| 3.8.6 – ESTRUTURA TIPO “ROM” .....                          | 90         |
| 3.9 – SINCRONISMO DE BITS (“BIT SYNC”).....                 | 90         |
| 3.9.1 – SINAIS DE CONTROLE .....                            | 90         |
| 3.9.2 – SINCRONIZAÇÃO DOS BITS .....                        | 95         |
| 3.10 – PORTA DE SAÍDA.....                                  | 97         |
| 3.11 – RESULTADOS OBTIDOS EM SIMULAÇÃO.....                 | 98         |
| 3.11.1 – CARACTERÍSTICAS GERAIS.....                        | 98         |
| 3.11.2 – CONSUMO DE POTÊNCIA.....                           | 99         |
| 3.11.3 – ERRO DE QUANTIZAÇÃO.....                           | 99         |
| 3.11.4 - RESULTADOS ESTÁTICOS .....                         | 99         |
| 3.11.5 - RESULTADOS DINÂMICOS .....                         | 100        |
| 3.11.5.1 – NÚMERO EFETIVO DE BITS .....                     | 100        |
| 3.11.5.2 – TESTE DA FREQUÊNCIA DE BATIMENTO.....            | 100        |
| 3.11.6 – “MONTE CARLO” .....                                | 101        |
| 3.11.7 – CORREÇÃO DE ERRO DE CÓDIGO .....                   | 102        |
| 3.11.8 – ANÁLISE DOS RESULTADOS .....                       | 103        |
| 3.11.9 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D .....       | 106        |
| 3.12 – “LAYOUT” .....                                       | 106        |
| <b>CAPÍTULO 4 - RESULTADOS.....</b>                         | <b>110</b> |
| 4.1 - AVALIAÇÃO DO CONVERSOR A/D .....                      | 110        |
| 4.1.1 – CONVERSORES A/D .....                               | 110        |
| 4.1.2 - LABORATÓRIO.....                                    | 111        |
| 4.2 - CONVERSOR A/D DE 6 BITS .....                         | 111        |
| 4.2.1 – DESCRIÇÃO.....                                      | 111        |
| 4.2.2 – MICROGRAFIA DO CIRCUITO INTEGRADO .....             | 111        |
| 4.2.3 – SISTEMA DE TESTE .....                              | 112        |
| 4.2.4 – SINAL DE SAÍDA RECONSTRUÍDO.....                    | 113        |
| 4.2.5 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D - 6 BITS.... | 113        |
| 4.3 - CONVERSOR A/D DE 8 BITS (TOPOLOGIA CONVENCIONAL)..... | 114        |
| 4.3.1 – DESCRIÇÃO.....                                      | 114        |
| 4.3.2 – MICROGRAFIA DO CIRCUITO INTEGRADO .....             | 114        |
| 4.3.3 – SISTEMA DE TESTE .....                              | 115        |
| 4.3.4 – CONSUMO DE POTÊNCIA .....                           | 117        |
| 4.3.5 – SINAL DE SAÍDA RECONSTRUÍDO.....                    | 117        |
| 4.3.6 – TESTE DA FREQUÊNCIA DE BATIMENTO.....               | 118        |
| 4.3.7 – ERRO DE QUANTIZAÇÃO.....                            | 120        |
| 4.3.8 – RESULTADOS ESTÁTICOS.....                           | 120        |
| 4.3.9 – RESULTADOS DINÂMICOS.....                           | 121        |
| 4.3.10 – ANÁLISE DOS CINCO BITS MENOS SIGNIFICATIVOS.....   | 122        |
| 4.3.11 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D - 8 BITS..  | 123        |
| 4.4 - CONVERSOR A/D DE 8 BITS (COM A NOVA TOPOLOGIA).....   | 123        |
| 4.4.1 – DESCRIÇÃO.....                                      | 123        |
| 4.4.2 – MICROGRAFIA DO CIRCUITO INTEGRADO .....             | 124        |
| 4.4.3 – ERRO NO “LAYOUT” .....                              | 125        |
| 4.4.4 – SISTEMA DE TESTE .....                              | 126        |
| 4.4.5 – CONSUMO DE POTÊNCIA .....                           | 127        |
| 4.4.6 – SINAL DE SAÍDA RECONSTRUÍDO.....                    | 127        |

|                                                                |            |
|----------------------------------------------------------------|------------|
| 4.4.7 – ERRO DE QUANTIZAÇÃO.....                               | 127        |
| 4.4.8 – RESULTADOS ESTÁTICOS.....                              | 128        |
| 4.4.9 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D DE 8 BITS ..... | 129        |
| 4.5 – ANÁLISE DOS RESULTADOS .....                             | 129        |
| 4.5.1 - INTERPOLAÇÃO .....                                     | 129        |
| 4.5.2 – CORREÇÃO DE ERRO DE CÓDIGO .....                       | 131        |
| 4.5.3 – NOVA TOPOLOGIA .....                                   | 131        |
| <b>CONCLUSÕES FINAIS.....</b>                                  | <b>132</b> |
| <b>REFERÊNCIA BIBLIOGRÁFICA .....</b>                          | <b>134</b> |
| <b>BIBLIOGRAFIA CONSULTADA .....</b>                           | <b>136</b> |
| <b>ANEXO 1 – DETALHES DO “LAYOUT” .....</b>                    | <b>139</b> |
| <b>ANEXO 2 – AVALIAÇÃO DE CONVERSORES A/D .....</b>            | <b>154</b> |

# LISTA DE FIGURAS

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| FIGURA I.1 – ARQUITETURA “FLASH” .....                                       | 1  |
| FIGURA I.2 - ARQUITETURA “FOLDING” .....                                     | 2  |
| FIGURA 1.1 - A TÉCNICA “FOLDING” .....                                       | 7  |
| FIGURA 1.2 - CDP E SUA CURVA DE TRANSFERÊNCIA .....                          | 8  |
| FIGURA 1.3 - EXEMPLO DE CIRCUITO DO AMPLIFICADOR “FOLDING” .....             | 8  |
| FIGURA 1.4 - VREF .....                                                      | 10 |
| FIGURA 1.5 - SINAIS DE SAÍDA DOS AMPLIFICADORES “FOLDING” .....              | 11 |
| FIGURA 1.6 - CIRCUITO BÁSICO DO “FOLDING ENCODER” .....                      | 13 |
| FIGURA 1.7 - SINAIS DE SAÍDA DO “FOLDING ENCODER” .....                      | 14 |
| FIGURA 1.8 - DIAGRAMA DE FASE 1 .....                                        | 14 |
| FIGURA 1.9 - INTERPOLAÇÃO BÁSICA .....                                       | 15 |
| FIGURA 1.10 - DIAGRAMA DE FASE 2 .....                                       | 15 |
| FIGURA 1.11 - SINAIS DIFERENCIAIS .....                                      | 16 |
| FIGURA 1.12 - DIAGRAMA DE FASE 3 .....                                       | 17 |
| FIGURA 1.13 - “LATCH” BÁSICO .....                                           | 18 |
| FIGURA 1.14 - MSB E 2SB .....                                                | 19 |
| FIGURA 1.15 - 3SB E S90 .....                                                | 20 |
| FIGURA 1.16 – SINAIS 3SB, S90 E INTERPOLADOS .....                           | 21 |
| FIGURA 1.17 - CORREÇÃO ANALÓGICA DE ERRO PROPOSTA RUDY VAN DE PLASSCHE ..... | 25 |
| FIGURA 1.18 - PORTA OU-EXCLUSIVO .....                                       | 26 |
| FIGURA 1.19 - SINAIS NA SAÍDA DAS PORTAS OU-EXCLUSIVO .....                  | 26 |
| FIGURA 1.20 – ESTRUTURA TIPO ROM .....                                       | 28 |
| FIGURA 1.21 – SINAIS DE SAÍDA DA ESTRUTURA TIPO ROM .....                    | 29 |
| FIGURA 1.22 - DESALINHAMENTO DOS “LATCHES” .....                             | 29 |
| FIGURA 1.23 - REGIÕES DE TRANSIÇÃO .....                                     | 30 |
| FIGURA 1.24 - CIRCUITOS LÓGICOS DE CORREÇÃO DE SINCRONISMO .....             | 31 |
| FIGURA 1.25 - PORTA DE SAÍDA .....                                           | 32 |
| FIGURA 1.26 - SINAIS DE SAÍDA .....                                          | 32 |
| FIGURA 2.1 - DISTORÇÃO NA INTERPOLAÇÃO .....                                 | 33 |
| FIGURA 2.2 - INTERPOLAÇÃO BÁSICA .....                                       | 34 |
| FIGURA 2.3 - DIAGRAMA DE FASE 1 .....                                        | 35 |
| FIGURA 2.4 - IMPLEMENTAÇÃO DA INTERPOLAÇÃO .....                             | 35 |
| FIGURA 2.5 - REDE RESISTIVA DE INTERPOLAÇÃO .....                            | 36 |
| FIGURA 2.6 - SINAIS NA REDE RESISTIVA DE INTERPOLAÇÃO .....                  | 37 |
| FIGURA 2.7 - ERRO “DC” NA REDE RESISTIVA DE INTERPOLAÇÃO .....               | 38 |
| FIGURA 2.8 - INTERPOLAÇÃO EM MODO DE CORRENTE .....                          | 39 |
| FIGURA 2.9 - IMPLEMENTAÇÃO DA INTERPOLAÇÃO EM MODO DE CORRENTE .....         | 40 |
| FIGURA 2.10 - SINAIS DIFERENCIAIS DE UM CONVERSOR A/D DE 6 BITS .....        | 42 |
| FIGURA 2.11 - INTERPOLAÇÃO SIMPLES .....                                     | 43 |
| FIGURA 2.12 - INTERPOLAÇÃO DUPLA .....                                       | 45 |
| FIGURA 2.13 - DISTORÇÃO .....                                                | 47 |
| FIGURA 2.14 - ASSIMETRIA DO SINAL .....                                      | 48 |
| FIGURA 2.15 – NÍVEL DO SINAL .....                                           | 48 |
| FIGURA 2.16 - SINAIS DIFERENCIAIS DE UM CONVERSOR A/D DE 8 BITS .....        | 50 |

|                                                                                                                                          |    |
|------------------------------------------------------------------------------------------------------------------------------------------|----|
| FIGURA 2.17 - SINAIS DE SAÍDA DOS AMPLIFICADORES “FOLDING” .....                                                                         | 51 |
| FIGURA 2.18 - CIRCUITO NO “FOLDING ENCODER” P/ OS SINAIS S45/ $\overline{S45}$ E S135/ $\overline{S135}$ .....                           | 52 |
| FIGURA 2.19 - SINAL S45/ $\overline{S45}$ .....                                                                                          | 53 |
| FIGURA 2.20 - SINAL S135/ $\overline{S135}$ .....                                                                                        | 53 |
| FIGURA 2.21 - SINAIS (S45 - $\overline{S45}$ ), (S135 - $\overline{S135}$ ), (3SB - $\overline{3SB}$ ) E (S90 - $\overline{S90}$ ) ..... | 54 |
| FIGURA 2.22 - VARIAÇÃO DOS SINAIS COM A TEMPERATURA .....                                                                                | 54 |
| FIGURA 2.23 - INTERPOLAÇÃO DUPLA .....                                                                                                   | 55 |
| FIGURA 2.24 - VARIAÇÃO, COM A TEMPERATURA, DOS SINAIS INTERPOLADOS .....                                                                 | 59 |
| FIGURA 2.25 - VARIAÇÃO, COM A TEMPERATURA, DA FASE DOS SINAIS INTERPOL. ....                                                             | 59 |
| FIGURA 2.26 - DISTORÇÃO NOS SINAIS INTERPOLADOS .....                                                                                    | 60 |
| FIGURA 2.27 - EXEMPLO DA SIMETRIA NECESSÁRIA NOS SINAIS INTERPOLADOS .....                                                               | 61 |
| FIGURA 3.1 - DIAGRAMA DE BLOCOS DO CONVERSOR A/D DE 8 BITS .....                                                                         | 63 |
| FIGURA 3.2 - CIRCUITO DO CONVERSOR A/D DE 8 BITS .....                                                                                   | 64 |
| FIGURA 3.3 - CDP (“COUPLED DIFFERENTIAL PAIR”) .....                                                                                     | 65 |
| FIGURA 3.4 - AMPLIFICADORES “FOLDING” .....                                                                                              | 65 |
| FIGURA 3.5 - CIRCUITO DE REFERÊNCIA DE VOLTAGEM .....                                                                                    | 65 |
| FIGURA 3.6 - SINAL DE ENTRADA DO CONVERSOR A/D ( $V_{IN}$ ) .....                                                                        | 66 |
| FIGURA 3.7 - SINAIS DE SAÍDA DOS AMPLIFICADORES “FOLDING” .....                                                                          | 66 |
| FIGURA 3.8 - CIRCUITO DO “FOLDING ENCODER” .....                                                                                         | 67 |
| FIGURA 3.9 - SINAIS DE SAÍDA DO “FOLDING ENCODER” .....                                                                                  | 68 |
| FIGURA 3.10 - FONTE DE POLARIZAÇÃO .....                                                                                                 | 69 |
| FIGURA 3.11 (A) – SEM FONTE PTAT .....                                                                                                   | 70 |
| FIGURA 3.11 (B) – COM FONTE PTAT .....                                                                                                   | 70 |
| FIGURA 3.12 - VARIAÇÃO DE $I_{FOLDING}$ COM A TEMPERATURA .....                                                                          | 70 |
| FIGURA 3.13 - CIRCUITO DE ENTRADA DO SINAL DE “CLOCK” .....                                                                              | 71 |
| FIGURA 3.14 - “LATCH” MESTRE SIMPLES .....                                                                                               | 72 |
| FIGURA 3.15 - SINAIS NA SAÍDA DOS “LATCHES” MESTRE SIMPLES .....                                                                         | 72 |
| FIGURA 3.16 - “LATCH” MESTRE DUPLO DE INTERPOLAÇÃO .....                                                                                 | 73 |
| FIGURA 3.17 - FASE DOS SINAIS NA SAÍDA DOS “LATCHES” MESTRE DUPLO DE INTERP. ....                                                        | 74 |
| FIGURA 3.18 - SINAIS NA SAÍDA DOS “LATCHES” MESTRE DUPLO DE INTERPOLAÇÃO ....                                                            | 76 |
| FIGURA 3.19 - GERAÇÃO DO 4SB .....                                                                                                       | 78 |
| FIGURA 3.20 - DIAGRAMA DE FASE DOS SINAIS RESULTANTES .....                                                                              | 79 |
| FIGURA 3.21 - SINAIS RESUL. NA SAÍDA DAS PORTAS “OU-EXCLUSIVO” COM “LATCH” .....                                                         | 81 |
| FIGURA 3.22 - PORTA “OU-EXCLUSIVO” COM “LATCH” .....                                                                                     | 81 |
| FIGURA 3.23 - SINAIS NA SAÍDA DAS PORTAS “OU-EXCLUSIVO” .....                                                                            | 83 |
| FIGURA 3.24 - CIRCUITO LÓGICO DE CORREÇÃO .....                                                                                          | 85 |
| FIGURA 3.25 - CIRCUITO DE CORREÇÃO .....                                                                                                 | 85 |
| FIGURA 3.26 - CÓDIGO CIRCULAR E O CIRCUITO DE CORREÇÃO .....                                                                             | 86 |
| FIGURA 3.27 - CORREÇÃO DO “EXTRA BIT” .....                                                                                              | 88 |
| FIGURA 3.28 - PORTA AND COM DUAS ENTRADAS .....                                                                                          | 89 |
| FIGURA 3.29 - PORTA AND COM TRÊS ENTRADAS .....                                                                                          | 89 |
| FIGURA 3.30 – ESTRUTURA TIPO ROM .....                                                                                                   | 90 |
| FIGURA 3.31 - SINAIS DE SAÍDA DA ESTRUTURA TIPO ROM .....                                                                                | 91 |
| FIGURA 3.32 - REGIÕES DE TRANSIÇÃO .....                                                                                                 | 91 |
| FIGURA 3.33 - CIRCUITO LÓG. PARA OBT. DOS SINAIS DE CONTROLE DO .....                                                                    | 93 |
| FIGURA 3.34 - SINAIS DE CONTROLE .....                                                                                                   | 94 |
| FIGURA 3.35 - CIRCUITO LÓGICO DE CORREÇÃO DO MSB, 2SB E 3SB .....                                                                        | 95 |

|                                                                                                                                     |     |
|-------------------------------------------------------------------------------------------------------------------------------------|-----|
| FIGURA 3.36 - SINAIS $MSB/\overline{MSB}$ , $2SB/\overline{2SB}$ , $3SB/\overline{3SB}$ , $4SB/\overline{4SB}$ E OS SIN. DE CONT... | 95  |
| FIGURA 3.37 - PORTA DE SAÍDA DO CONVERSOR A/D DE 8 BITS.....                                                                        | 97  |
| FIGURA 3.38 - SINAIS DE SAÍDA.....                                                                                                  | 98  |
| FIGURA 3.39 - CONVERSOR A/D E D/A INTERLIGADOS.....                                                                                 | 99  |
| FIGURA 3.40 – ERRO DE QUANTIZAÇÃO.....                                                                                              | 99  |
| FIGURA 3.41(a) – DNL.....                                                                                                           | 100 |
| FIGURA 3.41(b) – INL.....                                                                                                           | 100 |
| FIGURA 3.42 – NÚMERO EFETIVO DE BITS.....                                                                                           | 100 |
| FIGURA 3.43 – TESTE DA FREQUÊNCIA DE BATIMENTO.....                                                                                 | 100 |
| FIGURA 3.44 – CORREÇÃO DE ERRO.....                                                                                                 | 103 |
| FIGURA 3.45 - CONFIGURAÇÃO BÁSICA DO TRANSISTOR BIPOLAR.....                                                                        | 108 |
| FIGURA 3.46 - CONFIGURAÇÃO BÁSICA DE UM RESISTOR.....                                                                               | 108 |
| FIGURA 3.47 - “LAYOUT” FINAL DO CONVERSOR A/D DE 8 BITS.....                                                                        | 109 |
| FIGURA 4.1 – LABORATÓRIO DE INSTRUMENTAÇÃO E SENSORES - LIS.....                                                                    | 110 |
| FIGURA 4.2 – MICROGRAFIA DO CONVERSOR A/D DE 6 BITS.....                                                                            | 111 |
| FIGURA 4.3 – SISTEMA DE TESTE.....                                                                                                  | 112 |
| FIGURA 4.4 – EXEMPLO DE SINAL DE SAÍDA DO CONVERSOR A/D DE 6 BITS.....                                                              | 113 |
| FIGURA 4.5 – MICROGRAFIA DO CONVERSOR A/D 8 BITS (TOP. CONVENCIONAL).....                                                           | 115 |
| FIGURA 4.6 – SISTEMA DE TESTE.....                                                                                                  | 115 |
| FIGURA 4.7 (A) – CIRCUITO DA PLACA UTILIZADA NOS TESTES.....                                                                        | 116 |
| FIGURA 4.7 (B) – PLACA UTILIZADA NOS TESTES.....                                                                                    | 116 |
| FIGURA 4.8 – SINAL DE SAÍDA RECONSTRUÍDO.....                                                                                       | 117 |
| FIGURA 4.9(a) – SINAL COM “GLICHES”.....                                                                                            | 118 |
| FIGURA 4.9(b) – SINAL SEM “GLICHES”.....                                                                                            | 118 |
| FIGURA 4.10 – TESTE DA FREQUÊNCIA DE BATIMENTO.....                                                                                 | 119 |
| FIGURA 4.11(a) – SINAL COM “GLICHES”.....                                                                                           | 119 |
| FIGURA 4.11(b) – SINAL SEM “GLICHES”.....                                                                                           | 119 |
| FIGURA 4.12 – ERRO DE QUANTIZAÇÃO.....                                                                                              | 120 |
| FIGURA 4.13(a) – DNL.....                                                                                                           | 120 |
| FIGURA 4.13(b) – INL.....                                                                                                           | 120 |
| FIGURA 4.14(a) – NEB EM FUNÇÃO DE $F_{VIN}$ .....                                                                                   | 121 |
| FIGURA 4.14(b) – NEB EM FUNÇÃO DE $F_{CLOCK}$ .....                                                                                 | 121 |
| FIGURA 4.15 – SINAL RECONSTRUÍDO DOS CINCO BITS MENOS SIGNIFICATIVOS.....                                                           | 122 |
| FIGURA 4.16(a) – DNL.....                                                                                                           | 123 |
| FIGURA 4.16(b) – INL.....                                                                                                           | 123 |
| FIGURA 4.17 – MICROGRAFIA DO CONVERSOR A/D DE 8 BITS (NOVA TOPOLOGIA).....                                                          | 124 |
| FIGURA 4.18(a) – SAÍDA MEDIDA (REAL).....                                                                                           | 125 |
| FIGURA 4.18(b) – SAÍDA SIMULADA.....                                                                                                | 125 |
| FIGURA 4.19 – SISTEMA DE TESTE.....                                                                                                 | 126 |
| FIGURA 4.20(a) – SINAL DE SAÍDA RECONSTITUÍDO.....                                                                                  | 127 |
| FIGURA 4.20(b) – CINCO ÚLTIMOS BITS.....                                                                                            | 127 |
| FIGURA 4.21 – ERRO DE QUANTIZAÇÃO.....                                                                                              | 128 |
| FIGURA 4.22(a) – DNL.....                                                                                                           | 128 |
| FIGURA 4.22(b) – INL.....                                                                                                           | 128 |
| FIGURA 4.23(a) – SINAIS.....                                                                                                        | 129 |
| FIGURA 4.23(b) – INTERPOLAÇÃO.....                                                                                                  | 129 |
| FIGURA 4.24 – INTERPOLAÇÃO: RAZÃO DE ÁREAS E ÂNGULOS RESULTANTES.....                                                               | 131 |

# LISTA DE TABELAS

|                                                                                            |     |
|--------------------------------------------------------------------------------------------|-----|
| TABELA I.1 - EXEMPLOS DE CÓDIGOS.....                                                      | 2   |
| TABELA 1.1 - COMPLEXIDADE DO AMPLIFICADOR “FOLDING” .....                                  | 10  |
| TABELA 1.2 - CÓDIGO CIRCULAR .....                                                         | 22  |
| TABELA 1.3 - EXEMPLOS DOS CÓDIGOS .....                                                    | 23  |
| TABELA 1.4 - EXEMPLOS DE ERROS DE BOLHA.....                                               | 24  |
| TABELA 1.5 - SAÍDAS DAS PORTAS OU-EXCLUSIVO.....                                           | 27  |
| TABELA 1.6 - CORREÇÃO DE ERRO .....                                                        | 30  |
| TABELA 2.1 - IMPEDÂNCIAS DE SAÍDAS .....                                                   | 37  |
| TABELA 2.2 – ÂNGULOS X LSB .....                                                           | 49  |
| TABELA 2.3 - SINAIS APLICADOS E OS SINAIS GERADOS POR INTERPOLAÇÃO .....                   | 56  |
| TABELA 2.4 - FASE DOS SINAIS APLICADOS E AS RELAÇÕES DE $I_2/I_1$ OU $A(Q_2)/A(Q_1)$ ..... | 58  |
| TABELA 3.1 - SINAIS APLICADOS E OS SINAIS GERADOS POR INTERPOLAÇÃO .....                   | 75  |
| TABELA 3.2 - CÓDIGO CIRCULAR .....                                                         | 77  |
| TABELA 3.3 - SINAIS APLICADOS E OS SINAIS RESULTANTES .....                                | 79  |
| TABELA 3.4 - CÓDIGO CIRCULAR RESULTANTE.....                                               | 80  |
| TABELA 3.5 - SAÍDAS DAS PORTA OU-EXCLUSIVO .....                                           | 82  |
| TABELA 3.6 - EXEMPLOS DE CORREÇÃO DE ERRO .....                                            | 86  |
| TABELA 3.7 - CORREÇÃO DE ERRO DE SINCRONISMO .....                                         | 92  |
| TABELA 3.8 – CONSUMO DE POTÊNCIA.....                                                      | 99  |
| TABELA 3.9 – RESULTADOS ESTÁTICOS.....                                                     | 99  |
| TABELA 3.10 – RESULTADOS DINÂMICOS.....                                                    | 100 |
| TABELA 3.11 – RESULTADOS DA SIMULAÇÃO “MONTE CARLO” .....                                  | 101 |
| TABELA 3.12 – CORREÇÃO DE ERRO DE CÓDIGO.....                                              | 102 |
| TABELA 3.13 – ESPECIFICAÇÕES DO CONVERSOR A/D (SIMULADO) .....                             | 106 |
| TABELA 3.14 - RELAÇÃO BÁSICA DAS “LAYERS” DO PROCESSO .....                                | 107 |
| TABELA 3.15 - PARÂMETROS DE PROCESSO DO TRANSISTOR BIPOLAR.....                            | 107 |
| TABELA 4.1 – RELAÇÃO DOS EQUIPAMENTOS E SOFTWARE UTILIZADOS .....                          | 112 |
| TABELA 4.2 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D DE 6 BITS .....                        | 114 |
| TABELA 4.3 – RELAÇÃO DOS EQUIPAMENTOS E SOFTWARE UTILIZADOS .....                          | 116 |
| TABELA 4.4 – CONSUMO DE POTÊNCIA.....                                                      | 117 |
| TABELA 4.5 – RESULTADOS ESTÁTICOS.....                                                     | 121 |
| TABELA 4.6 – RESULTADOS DINÂMICOS.....                                                     | 121 |
| TABELA 4.7 – RESULTADOS ESTÁTICOS (LSB’s).....                                             | 122 |
| TABELA 4.8 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D DE 8 BITS.....                         | 123 |
| TABELA 4.9 – TROCA NOS SINAIS DE CONTROLE.....                                             | 125 |
| TABELA 4.10 – RELAÇÃO DOS EQUIPAMENTOS E SOFTWARE UTILIZADOS .....                         | 126 |
| TABELA 4.11 – CONSUMO DE POTÊNCIA.....                                                     | 127 |
| TABELA 4.12 – RESULTADOS ESTÁTICOS.....                                                    | 128 |
| TABELA 4.13 – ESPECIFICAÇÕES GERAIS DO CONVERSOR A/D DE 8 BITS .....                       | 129 |
| TABELA 4.14 – COMPARAÇÃO ENTRE AS TOPOLOGIAS.....                                          | 131 |

## LISTA DE ABREVIATURAS

A/D = analógico-digital  
ADC = conversor analógico-digital  
AMS = Austria Mikro Systeme International AG  
 $\overline{A-n}$  = sinal gerado no “folding encoder” ou por interpolação  
 $\overline{A-n}$  = sinal complementar ao A-n gerado no “folding encoder” ou por interpolação  
CDP = par diferencial acoplado (“coupled differential pair”)  
D/A = digital-analógico  
DAC = conversor digital-analógico  
DNL = não linearidade diferencial (“differential nonlinearity”)  
ECL = lógica acoplada por emissor (“emitter coupled logic”)  
 $f_T$  = frequência de transição  
I = corrente analógica genérica  
 $I_C$  = corrente do coletor de um transistor bipolar  
 $I_B$  = corrente da base de um transistor bipolar  
 $I_E$  = corrente do emissor de um transistor bipolar  
INL = não linearidade integral (“integral nonlinearity”)  
IS = corrente de saturação, como parâmetro do modelo de transistor bipolar  
LSB = bit menos significativo  
MSB = bits mais significativo  
N = número de bits de um conversor analógico-digital  
N' = número efetivo de bits de um conversor analógico-digital  
PTAT = proporcional a temperatura absoluta (“proportional to absolute temperature”)  
Q = referência a transistor bipolar  
R = resistor  
 $R_C$  = resistor de coletor do transistor bipolar  
 $R_B$  = resistor de base do transistor bipolar  
 $R_E$  = resistor de emissor do transistor bipolar  
 $R_L$  = resistor de coletor do transistor bipolar  
 $\overline{S-n}$  = sinal gerado no “folding encoder” ou por interpolação  
 $\overline{S-n}$  = sinal complementar ao S-n gerado no “folding encoder” ou por interpolação  
T = referência a transistor bipolar  
 $V_{BE}$  = tensão base-emissor de um transistor bipolar  
 $V_{in}$  = sinal (em tensão) de entrada  
 $V_r$  = sinal (em tensão) reconstruído na saída de um conversor digital-analógico  
 $V_{rms}$  = Tensão quadrática média (“root mean square”)  
 $V_T$  = tensão termodinâmica ( $k.T/q$ )

# Introdução

A arquitetura mais conhecida para conversores analógico/digitais rápidos é a técnica “flash”. A Fig. I.1 mostra um diagrama de blocos do conversor A/D tipo “flash”. Uma rede de comparadores compara o sinal analógico de entrada com determinadas voltagens de referência. Quando um sinal (tipo rampa) é aplicado na entrada, os comparadores terão na saída o nível lógico “um” quando a voltagem do sinal de entrada for maior do que a voltagem de referência e terão na saída o nível lógico “zero” quando for menor. Um grupo de “latches”, sendo um “latch” para cada comparador, retém a informação quando ocorre o sinal do “clock”. As saídas dos “latches” são posteriormente codificadas no código digital de saída. A arquitetura “flash” mostra uma boa performance em conversão rápida entretanto; requer  $(2^N - 1)$  comparadores para obter uma resolução de N bits, o que dificulta a obtenção de uma alta resolução e manter, ao mesmo tempo, uma grande largura de banda, baixa dissipação de potência e pequena área de integração. As desvantagens desta implementação são conhecidas: muitos componentes, grande área de “chip” e alto consumo de potência.

Além disso, na arquitetura tipo “flash”, é fácil perceber que cada “latch” participa de apenas uma decisão que resulta em troca no código digital de saída. O uso destes “latches” é claramente ineficiente, pois na saída dos comparadores existe um código termômetro (veja a Tabela I.1) e somente a posição de transição entre “um” e “zero” contém informação relevante

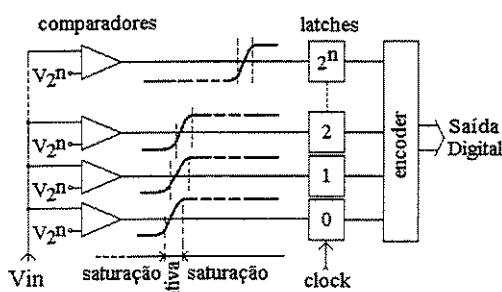


Figura I.1 – Arquitetura “Flash”

sobre o nível do sinal analógico de entrada. Embora seja uma prática comum nos conversores A/D tipo “flash” reter nos “latches” todos os níveis de saída dos comparadores, parte desta informação é redundante.

A técnica “folding” procura melhorar a

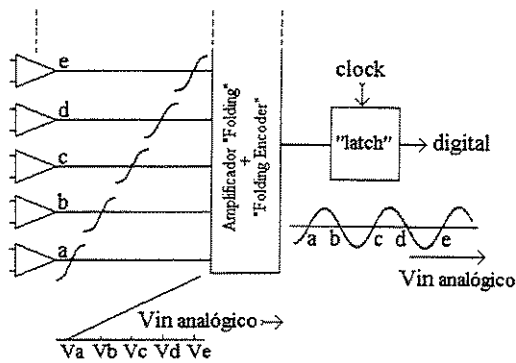


Figura I.2 – Arquitetura “Folding”

eficiência do conversor A/D utilizando um pré-processamento analógico (“folding” e interpolação). O decréscimo do número de “latches” é uma característica da técnica “folding”, em que poucos “latches” são necessários porque eles são utilizados de uma maneira mais eficiente. A Fig. I.2 mostra como o circuito de pré-processamento analógico foi inserido entre os comparadores e os “latches”. Os comparadores são conectados a este circuito. O amplificador “folding” mais o “folding encoder”, que fazem o pré-processamento analógico, dobram (“folds”) ou juntam os sinais de saída dos comparadores para alimentar um único “latch”. Desta forma, o “latch” recebe o sinal de saída de mais de um comparador e processa mais de um nível de quantização. Os comparadores são combinados para serem conectados à referência de voltagens suficientemente distantes para evitar que os sinais de saída sofram interferência mútua. Por causa da ação “folding” os mesmos “latches” são utilizados repetidas vezes. Portanto, o nome “folding” vem do inglês “folds” que, neste caso, seria dobrar/juntar as saídas de vários comparadores em um único “latch” e, desta forma, reduzir a redundância de “latches” existentes na conversão tipo “flash” [05].

A arquitetura “folding” com interpolação é capaz de obter um desempenho semelhante ao

Tabela I.1 – Exemplos de Códigos

|   | Código Termômetro | Código Circular |
|---|-------------------|-----------------|
| 0 | 0 0 0 0 0 0 0     | 0 0 0 0         |
| 1 | 0 0 0 0 0 0 1     | 0 0 0 1         |
| 2 | 0 0 0 0 0 1 1     | 0 0 1 1         |
| 3 | 0 0 0 0 1 1 1     | 0 1 1 1         |
| 4 | 0 0 0 1 1 1 1     | 1 1 1 1         |
| 5 | 0 0 1 1 1 1 1     | 1 1 1 0         |
| 6 | 0 1 1 1 1 1 1     | 1 1 0 0         |
| 7 | 1 1 1 1 1 1 1     | 1 0 0 0         |

da arquitetura “flash” com menor consumo de potência e menor área de integração. A técnica de interpolação associada à técnica “folding” permite o aumento da resolução do conversor A/D sem aumentar a complexidade do amplificador “folding”.

A arquitetura “folding” pode ser considerada como uma arquitetura “two-step” porque, de forma semelhante aos conversores A/D tipo “two-step” ou “pipeline, a conversão do sinal analógico é dividida em duas etapas. No conversor A/D tipo “folding” a conversão do sinal analógico consiste em um estágio de conversão para os bits mais significativos (“coarse bits”) e outro estágio de conversão para os bits menos significativos (“fine bits”). As duas etapas de conversão são realizadas em paralelo [12], o que permite ao conversor A/D tipo “folding” trabalhar com a frequência máxima de sinal de “clock”, igualando ao que se obtém com um conversor A/D tipo “flash” e, da mesma forma, não necessitando utilizar um circuito “track-and-hold” na entrada. Entretanto, a frequência do sinal analógico de entrada é multiplicada durante o pré-processamento analógico no conversor A/D. O fator de multiplicação é chamado de razão de “folding” do conversor A/D. Isto resulta em um compromisso porque quanto maior for a razão de “folding”, menor será o número de “latches” utilizados mas, ao mesmo tempo, menor será a máxima frequência do sinal de entrada do conversor A/D [12].

A seguir é dado um resumo histórico da evolução da técnica “folding”.

- Em fevereiro de 1975, A. Abel e K. Kurtz publicaram o artigo “Fast ADC” [01], que descreve o que seria a primeira idéia de um conversor A/D tipo “folding”.
- Em junho de 1979, Udo Fiedler e Dieter Seitzer publicaram o artigo “A High Speed 8 Bit A/D Converter Based on a Gray-Code Multiple Folding Circuit” [02], no qual foi descrito um conversor A/D de 8 bits utilizando-se a técnica “folding” simples.
- Em dezembro de 1979, Rudy J. van de Plassche e Rob E. J. van der Grift publicaram o artigo “A High-Speed 7 Bit A/D Converter” [03], no qual foi descrito um conversor de 7 bits que utilizava a técnica “folding” simples. Rudy J. van de Plassche é professor da “Eindhoven University of Technology” e pesquisador do “Philips Research Laboratories” e tem sido o maior pesquisador da técnica “folding”.

- Em junho de 1984, Rob E. J. van der Grift e Rudy J. van de Plassche publicaram o artigo “A Monolithic 8-Bit Video A/D Converter” [04], no qual foi descrito o primeiro conversor A/D utilizando-se a técnica duplo “folding”.

- Em dezembro de 1987, Rob E. J. Van de Grift, Ivo W. J. M. Rutten e Martien Van der Veen publicaram o artigo “An 8 bit Video ADC Incorporating Folding and Interpolation Techniques” [05], no qual foi descrito um conversor A/D de 8 bits utilizando-se a técnica duplo “folding” com interpolação.

- Em dezembro de 1988, Rudy J. van de Plassche e Peter Baltus publicaram o artigo “An 8-bit 100-MHz Full-Nyquist Analog-to-Digital Converter” [06], no qual foi descrito um conversor de 8 bits e de grande desempenho, utilizando-se a técnica duplo “folding” com interpolação. Este artigo é importante pela descrição completa que faz do conversor A/D tipo “folding” com interpolação e do conceito de sincronismo de bits.

- Em dezembro de 1992, Johan van Valburg e Rudy J. van de Plassche publicaram o artigo “An 8-b 650-MHz Folding ADC” [07], no qual foi apresentado um conversor A/D de 8 bits de 650 MHz com uma técnica analógica de correção de erro de bolha no código circular interno.

- Em 1994, Rudy J. van de Plassche publicou o livro “Integrated Analog to Digital and Digital to Analog Converters” [08]. Este livro contém uma descrição completa da técnica “folding” e, mais especificamente, da técnica duplo “folding” com interpolação.

- Em dezembro de 1995, Bram Nauta e Ardie G. W. Venes publicaram o artigo “A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter” [09], no qual foi apresentado um conversor A/D tipo duplo “folding” com interpolação em tecnologia CMOS.

- Em outubro de 1995, C. K. Poulton, K. L. Knudsen e J. J. Corcoran publicaram o artigo “A 6b 4 GS/s GaAs HBT ADC” [10], no qual foi apresentado um conversor A/D tipo “folding”

ultra rápido em GaAs.

- Em setembro de 1996, Michael P. Flynn e David J. Allstot publicaram o artigo “CMOS Folding A/D Converters with Current-Mode Interpolation” [11], no qual foram apresentados dois conversores A/D de 6 e 8 bits, utilizando-se a técnica duplo “folding” com uma nova técnica de interpolação por divisão de corrente em tecnologia CMOS.

- Em dezembro de 1996, Ardie G. W. Venes e Rudy J. van de Plassche publicaram o artigo “An 80-MHz, 80-mW, 8-b CMOS Folding A/D Converter with Distributed Track-and-Hold Preprocessing” [12], no qual foi apresentado um conversor A/D de 8 bits, utilizando-se a técnica duplo “folding” com interpolação em tecnologia CMOS.

- Em novembro de 1997, R. Alini et al. publicaram o artigo “A 200 MSample/s Trellis-Coded PRML, Read/Write Channel with Analog Adaptive Equalizer and Digital Servo” [13], no qual foi apresentado um conversor A/D tipo “folding” com interpolação de 6 bits em tecnologia CMOS para uso em sistema de leitura e gravação em canais magnéticos.

- Em dezembro de 1997, Pieter Vorenkamp e Raf Roovers publicaram o artigo “A 12-b, 60-MSample/s Cascaded Folding and Interpolating ADC” [14], no qual foi apresentado, em processo 1  $\mu\text{m}$  BiCMOS (área do “chip” igual a 7  $\text{mm}^2$ ), um conversor A/D tipo “folding” cascadeado com interpolação de alta resolução (12 bits).

- Em dezembro de 1997, K. Bult e A. Buchwald publicaram o artigo “An Embedded 240-mW 10-b 50-MS/s CMOS ADC in 1- $\text{mm}^2$ ” [15], no qual foi apresentado, em processo 0,5  $\mu\text{m}$  CMOS (área do “chip” igual a 1  $\text{mm}^2$ ), um conversor A/D tipo “folding” cascadeado com interpolação de alta resolução (10 bits).

- Em dezembro de 1998, Michael P. Flynn e Ben Sheahan publicaram o artigo “A 400-MSample/s, 6-b CMOS Folding and Interpolating ADC” [16], no qual foi apresentado um conversor A/D tipo duplo “folding” com interpolação por divisão de corrente usando-se

tecnologia CMOS.

De forma geral, a arquitetura “folding” com interpolação é útil para conversores A/D rápidos com baixo consumo de potência e área relativamente pequena de integração, permitindo resolução de 6 a 12 bits em tecnologia bipolar e resolução de 6 a 10 bits em tecnologia CMOS. É dentro deste espaço que este trabalho se enquadra, tendo por objetivo propor um conversor A/D ultra rápido bipolar tipo duplo “folding” com uma nova técnica de interpolação, correção de erro no código circular interno e sincronismo de bits. O conversor A/D tipo “folding” com interpolação possui vantagens significativas para competir com sucesso no grupo de conversores A/D rápidos. Isto será demonstrado e percebido na proposta apresentada. O texto seguiu a terminologia e as definições adotadas em [08].

O capítulo 1 faz uma revisão geral da técnica duplo “folding” com interpolação. O capítulo 2 mostra as técnicas de interpolação existentes e a nova técnica proposta. O capítulo 3 mostra o projeto do conversor A/D com a nova técnica de interpolação, a correção de erro de código e uma nova topologia para o circuito de conversão. O capítulo 4 apresenta os resultados obtidos. E as conclusões finais fazem uma análise geral dos resultados e indicam a continuidade do trabalho.

# Capítulo 1

## A Técnica Duplo “Folding” com Interpolação

### 1.1 – Descrição da Técnica:

O sistema duplo “folding” foi desenvolvido como um aperfeiçoamento do sistema “folding” tradicional [08]. Um diagrama de blocos do conversor analógico/digital utilizando a técnica duplo “folding” com interpolação é mostrado na Fig. 1.1. No conversor A/D duplo “folding”, os circuitos duplo “folding” determinam diretamente os bits mais significativos (“coarse bits”) e os dois sinais analógicos com deslocamento de fase de  $90^\circ$ . Estes dois sinais analógicos são usados para interpolação e para se obter os demais sinais necessários para incrementar a resolução do conversor. Desta forma, os bits menos significativos (“fine bits”) são determinados pela apropriada combinação dos sinais interpolados [04, 05, 07 e 08]. Estes sinais

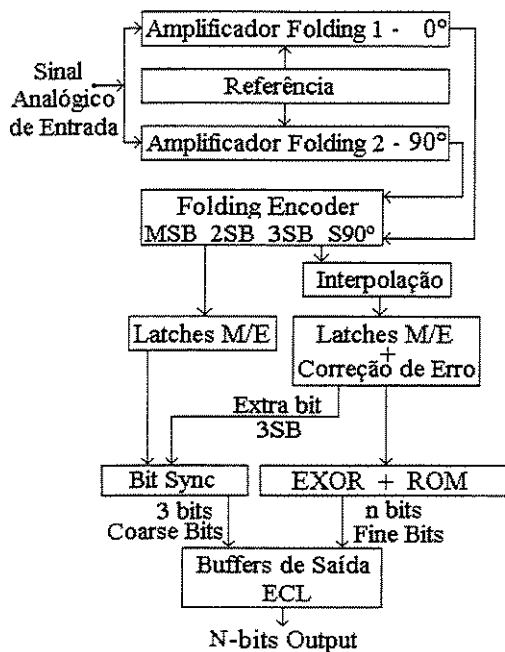


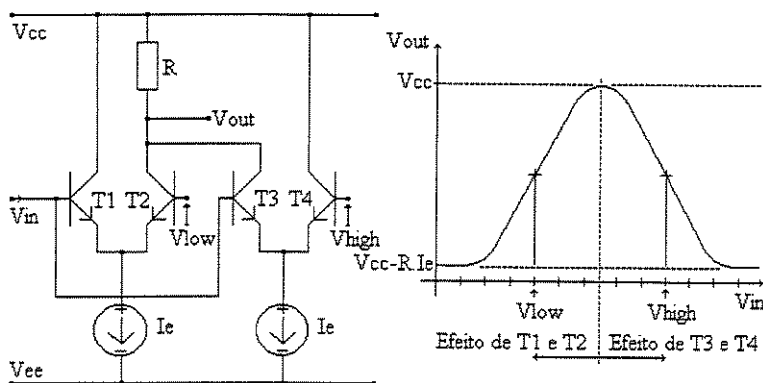
Figura 1.1 – A Técnica “Folding”

interpolados têm toda a informação necessária para determinar os bits menos significativos do conversor A/D. Um conjunto de “latches” mestre-escravo transforma a informação analógica interpolada em um código circular. Este código pode ser convertido em uma forma binária usando-se portas “ou-exclusivo” (EXOR) e uma estrutura codificadora binária tipo “ROM”. As saídas desta estrutura codificadora tipo “ROM” contêm os bits menos significativos que alimentam os “buffers” de saída “ECL”. A seguir será demonstrada a técnica duplo

“folding” com interpolação, tomando como base um conversor A/D com resolução de 8 bits.

## 1.2 – Amplificador “Folding”:

O amplificador “folding” é constituído de diversos estágios pré-amplificadores acoplados. A compreensão do funcionamento do amplificador “folding” está no entendimento deste acoplamento entre os estágios pré-amplificadores. Observando o circuito da Fig. 1.2, temos o chamado CDP (“Coupled Differential Pair”) que foi proposto em [07]. A curva de transferência do CDP pode ser vista na mesma figura. O primeiro par diferencial se referencia em  $V_{low}$  e, quando  $V_{in}$  fica maior do que  $V_{low}$ , a voltagem de saída do CDP ficará superior a “ $V_{cc}-R.I_e/2$ ”. O segundo par diferencial se referencia de modo complementar em  $V_{high}$  e, quando  $V_{in}$  fica maior do que  $V_{high}$ , a voltagem de saída do CDP ficará inferior à “ $V_{cc}-R.I_e/2$ ”. A diferença entre as duas voltagens  $V_{high}$



e  $V_{low}$  ( $\Delta V = V_{high} - V_{low}$ )

indica o intervalo em que a voltagem de saída do CDP ficará com valor igual ou acima de “ $V_{cc}-R.I_e/2$ ”.

Colocando “buffers”

em  $V_{in}$ ,  $V_{ref}$  e  $V_{out}$  e elimi-

nando as redundâncias dos diversos estágios pré-amplificadores acoplados, obtemos um

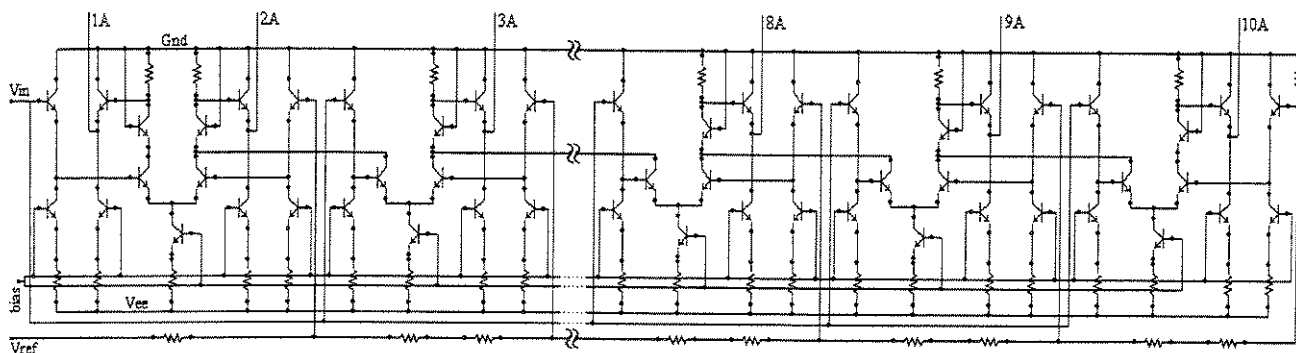


Figura 1.3 – Exemplo de Circuito do Amplificador “Folding”

amplificador “folding” completo com 10 saídas (veja a Fig. 1.3). A quantidade de estágios pré-amplificadores e voltagens de referências está diretamente relacionada e determina o número de bits mais significativos que será gerado diretamente pelo amplificador “folding”. Exemplificando: para gerar diretamente 3 bits, deve-se fazer uma combinação de 8 CDP’s (veja a Fig. 1.3) com a voltagem de referência dividida em 8 intervalos ( $V_{ref}/8$ ) e, para gerar diretamente 4 bits, deve-se fazer uma combinação de 16 CDP’s com a voltagem de referência dividida em 16 intervalos ( $V_{ref}/16$ ) e assim sucessivamente. Portanto, a cada aumento da resolução em 1 bit, o número de CDP’s dobra. O número de CDP’s é igual a  $2^N$ , sendo “N” o número de bits gerados diretamente no amplificador “folding”. Teoricamente seria possível gerar todos os bits do conversor A/D através da combinação das curvas de transferência dos CDP’s, porém, isto não é recomendável por dois motivos: Primeiro, o intervalo da voltagem de referência acompanha o número de bits do código digital, isto é, o intervalo corresponde a  $[1/(2^N)]$  da voltagem de referência. Em um amplificador “folding” de 3 bits, devem ser utilizados intervalos de  $[1/(2^3)] = 1/8$  da voltagem de referência; isto corresponde a 8 CDP’s. Se o amplificador “folding” fosse gerar diretamente 8 bits, seriam necessários intervalos de  $[1/(2^8)] = 1/256$  da voltagem de referência, o que resultaria em 256 CDP’s. A complexidade deste sistema seria semelhante à de um conversor A/D tipo “flash” de 8 bits, o que não é desejável.

Segundo, existe um limite prático para uso desta técnica por causa do aumento da frequência do sinal interno no amplificador “folding”. Por exemplo, para gerar diretamente 3 bits, foi necessário utilizar voltagens de referência divididas em 8 intervalos ( $V_{ref}/8$ ), o que origina uma frequência da ordem de 8 vezes a frequência do sinal de entrada ( $V_{in}$ ) no amplificador “folding”. Este fator, que neste exemplo é 8, chama-se razão de “folding” do conversor [07]. Neste caso, a razão de “folding” deste conversor seria 8. Se  $V_{in}$  tivesse uma frequência de 100 MHz, a frequência interna no amplificador “folding” chegaria a 800 MHz. Através da razão de

“folding”, é possível fixar o número aproximado de bits que podem ser facilmente (ou diretamente) derivados do sistema utilizando-se combinações das saídas dos CDP’s do amplificador “folding”, conforme [07]. Este número é dado pela seguinte relação:  $N = \log_2(\text{razão de “folding”})$ . Se, por exemplo, a razão de “folding” fosse fixada em 8, teríamos:  $\log_2(8) = 3$ . O valor obtido seria o número de bits (3) que poderiam ser diretamente gerados, combinando-se as saídas dos CDP’s e mantendo-se a razão de “folding” constante em oito (8).

Portanto, a largura de banda no amplificador “folding” ( $BW_{\text{fold}}$ ) será o fator limitante em relação à máxima frequência permitida no sinal analógico de entrada do conversor A/D. Isto ocorre porque o amplificador “folding” possui este efeito de multiplicador de frequência dependente do sinal de entrada [12]. Para um sinal senoidal de entrada, a máxima frequência permitida seria dada pela equação abaixo [12], onde  $F_r$  é a razão de

“folding”.

$$f_{\text{in,max}} = \frac{2 \cdot BW_{\text{fold}}}{\pi \cdot F_r} \quad (1.1)$$

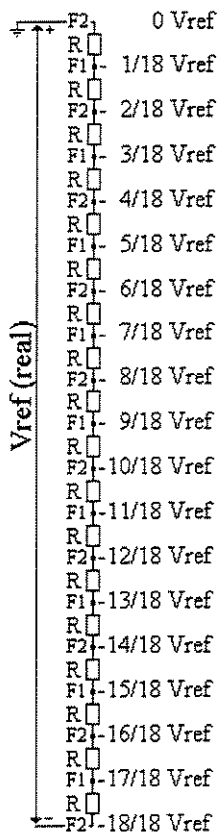


Tabela 1.1 - Complexidade do Amplificador “Folding”

| Número de bits gerados | Número de CDP’s | Intervalo de $V_{\text{ref}}$ | Razão de “Folding” |
|------------------------|-----------------|-------------------------------|--------------------|
| 2                      | 4               | 1/4                           | 4                  |
| 3                      | 8               | 1/8                           | 8                  |
| 4                      | 16              | 1/16                          | 16                 |
| 5                      | 32              | 1/32                          | 32                 |
| 8                      | 256             | 1/256                         | 256                |

A Tabela 1.1 mostra as relações básicas entre o número de bits diretamente gerados e a complexidade do amplificador “folding”. Na prática, é razoável utilizar uma razão de “folding” igual a 8 [08], gerando

Figura 1.4 -  $V_{\text{ref}}$

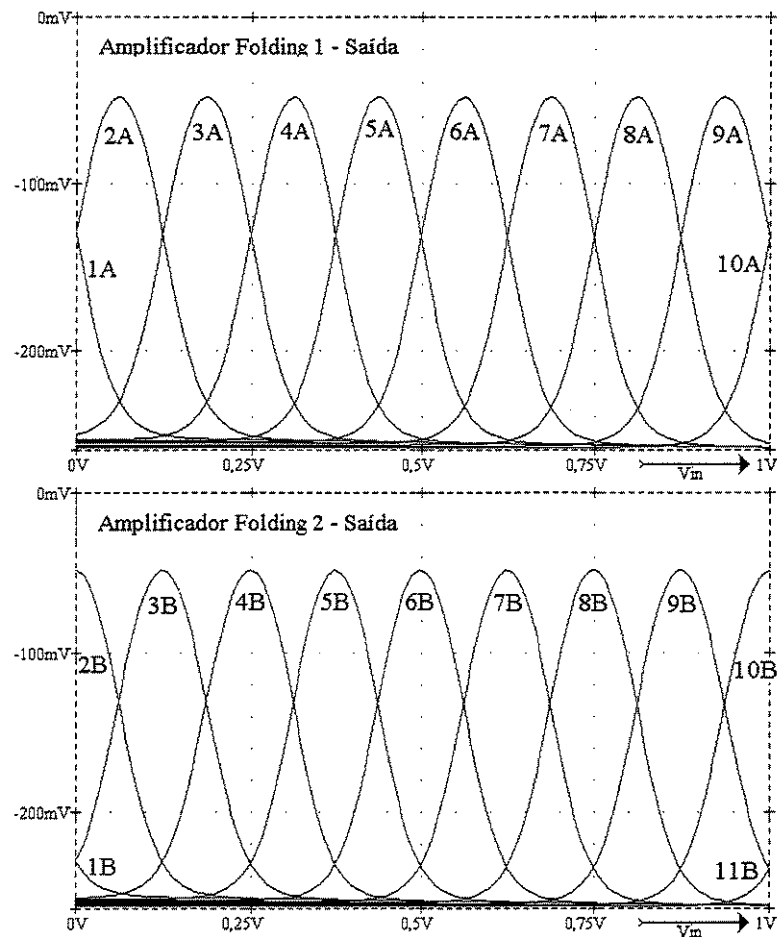


Figura 1.5 – Sinais de Saída dos Amplificadores “folding”

diretamente 3 bits e evitando o aumento da complexidade e da razão de “folding”. Na geração dos demais bits (considerando um conversor A/D de 8 bits), devem-se colocar dois amplificadores “folding” em paralelo (daí, o nome duplo “folding”) e utilizar a interpolação (descrita mais adiante) [05, 07 e 08]. A solução adotada na técnica duplo “folding” é exatamente colocar outro amplificador “folding” em paralelo com o primeiro e posteriormente utilizar a interpolação.

Alguns sinais de saída dos dois amplificadores “folding” após serem combinados no circuito chamado “folding encoder” tomam uma forma quase senoidal e com a mesma frequência. Estes sinais quase senoidais, de mesma frequência, possuem “diferenças de fase” que estão relacionadas com as diferenças nas voltagens de referência. Desta forma, a fase indica o quanto um sinal senoidal está deslocado em relação a outro (nestes caso:  $90^\circ = V_{ref}/2^{(N+1)} = V_{ref}/16$ ).

Desta forma, ao longo do texto, esta diferença de fase e gráficos de fase serão utilizados para visualizar a obtenção dos bits menos significativos.

O segundo amplificador “folding” deve ter a voltagem de referência deslocada de um “offset” de  $1/16 V$  (para uma razão de “folding” igual a 8) em relação à voltagem de referência do primeiro amplificador “folding”. Este “offset” implica que o sinal gerado pelo segundo amplificador “folding” tem uma fase de  $90^\circ$  em relação ao sinal gerado pelo primeiro amplificador “folding” (veja a Fig. 1.5). Sendo a voltagem de referência comum aos dois amplificadores “folding”, o “offset” pode ser facilmente implementado intercalando-se resistores na rede resistiva utilizada como referência, conforme pode ser visto na Fig. 1.4. Nesta figura, os pontos identificados por “F1” são utilizados como referência pelo amplificador “folding” 1 e os pontos identificados por “F2”, de forma semelhante, são utilizados como referência pelo amplificador “folding” 2. Na prática, utiliza-se  $V_{ref(real)} = 1,125.V_{ref(ideal)}$  e acrescentam-se mais dois resistores na rede resistiva utilizada como referência. Isto permite ajustar corretamente as formas iniciais das ondas geradas pelos dois amplificadores “folding” (1A, 10A, 1B, 2B, 10B e 11B). Desta forma, o sinal analógico de entrada dever ter um “offset”. Os sinais de saída dos dois amplificadores “folding” podem ser vistos na Fig. 1.5 (na realidade existem vários pontos de saída, um para cada CDP).

### 1.3 – “Folding Encoder”:

O circuito “folding encoder” é basicamente uma estrutura tipo “ROM” que combina os sinais de saída dos amplificadores “folding” em 8 sinais complementares (veja a Fig. 1.6) [06 e 08]. Estes sinais são a versão analógica do MSB,  $\overline{MSB}$ , 2SB,  $\overline{2SB}$ , 3SB,  $\overline{3SB}$ , S90 e  $\overline{S90}$  e se tornarão sinais digitais após passarem pelos “latches” mestre. Veja, a seguir, as combinações dos sinais das saídas do amplificador “folding”, no “folding encoder” e compare com as Figs. 1.5 e 1.7:

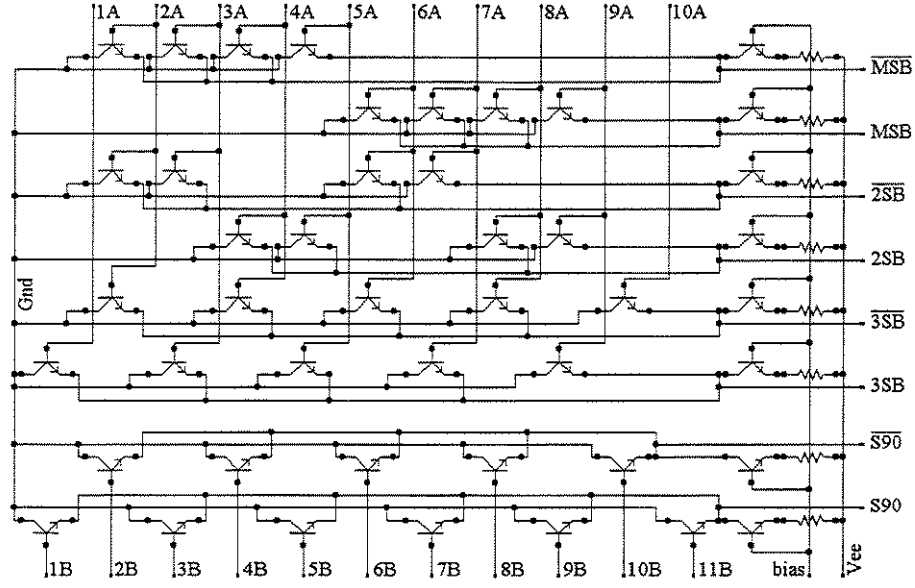


Figura 1.6 – Circuito Básico do “Folding Encoder”

$$\text{MSB} \Rightarrow (6A + 7A + 8A + 9A) \text{ e } \overline{\text{MSB}} \Rightarrow (2A + 3A + 4A + 5A)$$

$$2\text{SB} \Rightarrow (4A + 5A + 8A + 9A) \text{ e } \overline{2\text{SB}} \Rightarrow (2A + 3A + 6A + 7A)$$

$$3\text{SB} \Rightarrow (1A + 3A + 5A + 7A + 9A) \text{ e } \overline{3\text{SB}} \Rightarrow (2A + 4A + 6A + 8A + 10A)$$

$$\text{S90} \Rightarrow (1B + 3B + 5B + 7B + 9B + 11B) \text{ e } \overline{\text{S90}} \Rightarrow (2B + 4B + 6B + 8B + 10B)$$

A versão analógica dos sinais MSB,  $\overline{\text{MSB}}$ , 2SB,  $\overline{2\text{SB}}$ , 3SB e  $\overline{3\text{SB}}$  é obtida por combinação dos sinais oriundos do amplificador “folding” 1 e versão analógica dos sinais S90 e  $\overline{\text{S90}}$ , de forma semelhante, é obtida por combinação dos sinais oriundos do amplificador “folding” 2. A Fig. 1.7 mostra estes mesmos sinais na forma diferencial.

Na saída do “folding encoder” existe a versão analógica do bit MSB, do bit 2SB e do bit 3SB [12]. A ondulação observada sobre os sinais  $(\text{MSB} - \overline{\text{MSB}})$  e  $(2\text{SB} - \overline{2\text{SB}})$  é inerente a sua formação, ou seja, é o resultado da soma dos sinais do amplificador “folding” e não acarreta nenhum problema, porque irá desaparecer quando o sinal passar pelo “latches” mestre do conversor A/D. Os sinais  $(3\text{SB} - \overline{3\text{SB}})$  e  $(\text{S90} - \overline{\text{S90}})$  têm a mesma frequência e uma diferença de fase de 90°. O diagrama de fase destes dois sinais pode ser visto na Fig. 1.8.

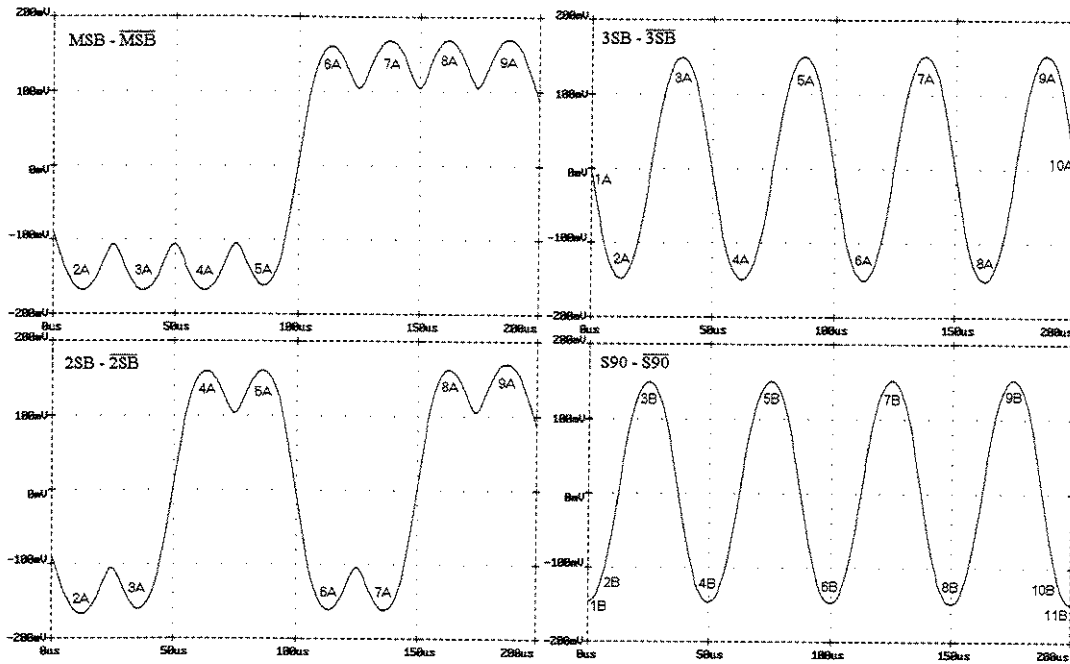


Figura 1.7 – Sinais de Saída do “Folding Encoder”

### 1.4 – Interpolação:

A técnica “folding” reduz o número de “latches”, mas parte da informação é perdida [16] e deve ser recuperada para a obtenção dos bits menos significativos. A recuperação desta informação perdida é possível através da técnica de interpolação entre dois sinais de saída do “folding encoder”.

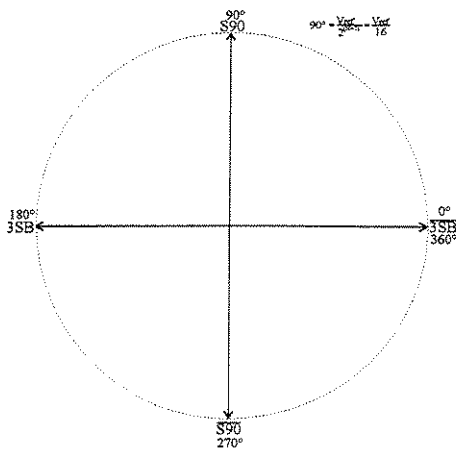
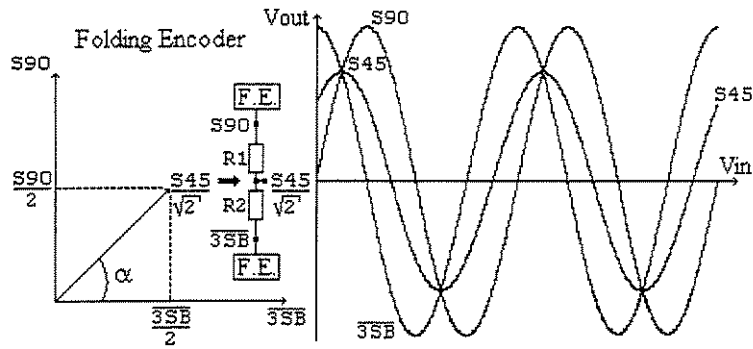


Figura 1.8 – Diagrama de Fase 1

As diversas formas de implementação da interpolação serão mostradas e analisadas detalhadamente no próximo capítulo. O princípio da interpolação é bastante simples e um exemplo básico da sua implementação pode ser entendido com a ajuda da Fig. 1.9. Teoricamente, a interpolação pode ser definida como uma soma entre dois sinais ( $\overline{3SB}$  e  $S90$ ), em que se considera a amplitude e a fase destes sinais a serem



Obs: F.E. = "Folding Encoder"

Figura 1.9 – Interpolação Básica

[ $R/(R+R)= 1/2$ ] e as parcelas que realmente participam da soma são  $3SB/2$  e  $S90/2$ , o que justifica a amplitude do sinal resultante.

É justamente no ajuste da relação da amplitude e fase dos sinais a serem somados que se obtém o sinal resultante na posição exata (ângulo) de interpolação desejada. Isto é, pode ser realizada uma ponderação relativa de cada sinal na obtenção do sinal resultante (S45). No exemplo da Fig. 1.9, o ângulo “ $\alpha$ ” do sinal resultante S45 terá  $45^\circ$  se os dois resistores ( $R_1$  e  $R_2$ ) forem iguais (veja a Fig. 1.10). Se os resistores forem diferentes, o ângulo “ $\alpha$ ” terá um valor diferente. A interpolação permite obter sinais com a mesma frequência dos sinais interpoladores e diferentes ângulos de fase. A amplitude do sinal resultante tem pouca importância porque todos

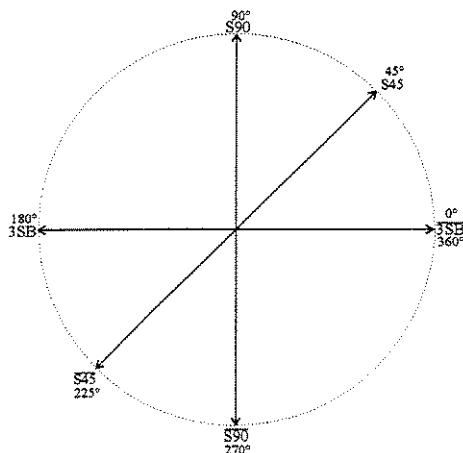


Figura 1.10 – Diagrama de Fase 2

somados e se obtém igualmente um sinal resultante (S45) com a sua própria amplitude e fase, conforme pode ser observado na Fig. 1.9. Supondo  $R_1 = R_2 = R$ , os sinais  $3SB$  e  $S90$  sofrem o efeito da divisão de voltagem

os sinais irão passar por “latches” e serão transformados em sinais digitais. De forma geral, é possível dizer que as fases dos sinais interpolados posicionam-se entre as fases dos sinais interpoladores, conforme a influência de cada um na interpolação. Na realidade, a interpolação equivale à colocação de amplificadores “folding” em paralelo

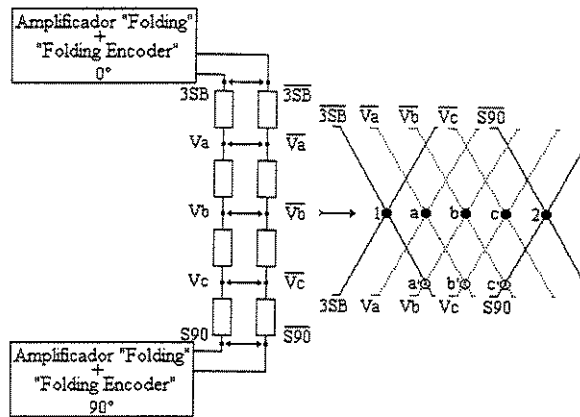


Figura 1.11 – Sinais Diferenciais

com o seu sinal de saída defasados de algumas frações da voltagem de referência (o que resulta na diferença de fase).

Na prática, a implementação da interpolação é mais aprimorada, sendo realizada de forma diferencial.

Na Fig. 1.11, observe que os sinais de saída ( $3SB / \overline{3SB}$ ,  $V_a / \overline{V_a}$ ,  $V_b / \overline{V_b}$ ,  $V_c / \overline{V_c}$  e  $S90 / \overline{S90}$ ) são diferenciais e o instante do cruzamento por zero é detectado pelo “latch” (com entrada diferencial) quando os sinais diferenciais (por exemplo:  $V_a$  e  $\overline{V_a}$ ) forem iguais. Na Fig. 1.11 é possível ver os pontos pretos marcados quando os sinais diferenciais se tornam iguais (os pontos correspondentes a “1”, “a”, “b”, “c” e “2”). Inclusive pode-se perceber que existem outros pontos marcados que também permitiriam identificar o mesmo instante. Por exemplo, o cruzamento dos sinais  $V_b$  e  $\overline{3SB}$  (o momento em que forem iguais é identificado pelo ponto a’), também indica o instante em que os sinais  $V_a$  e  $\overline{V_a}$  se tornam iguais (ponto “a”). Isto é muito útil porque permite simplificar e melhorar o circuito de interpolação, que não precisa trabalhar com sinais de referência fixa e sim com sinais diferenciais.

Após a análise anterior, é possível entender como a interpolação recupera a informação perdida. Considerando um conversor A/D de 8 bits, a saída do “folding encoder” já tem os 3 bits mais significativos. Para se obterem os 5 bits menos significativos são necessários mais 32 ( $2^5$ ) níveis de quantização, que correspondem a mais 32 sinais; sendo 2 já existentes (3SB e S90) e mais 30 a serem obtidos através da interpolação. Teoricamente, o número de sinais interpolados

poderia ser maior; porém, na prática, este tem sido o limite considerado para aumentar a resolução de um conversor A/D [06 e 08]. Estes 32 sinais estão defasados de  $5,625^\circ$  (equivalente a um  $\text{LSB} = V_{\text{ref}}/2^N = V_{\text{ref}}/2^8 = V_{\text{ref}}/256$ ) e contêm todas as informações necessárias para gerar os 5 bits menos significativos do conversor A/D. O diagrama de fase destes sinais pode ser visto na Fig. 1.12.

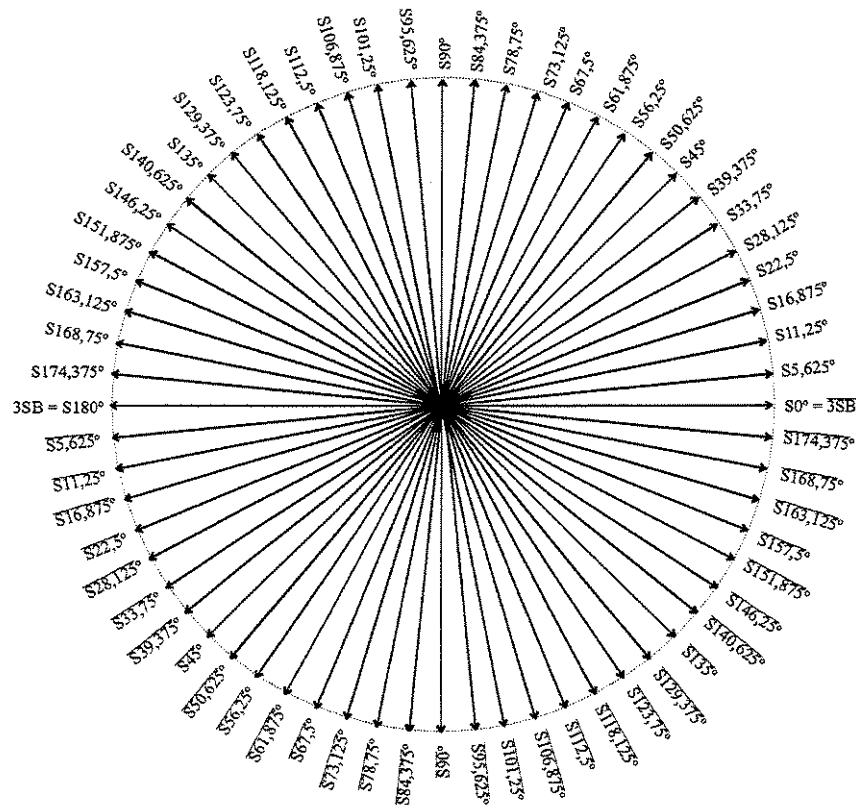


Figura 1.12 – Diagrama de Fase 3

### 1.5 – “Latches” Mestre-Escravo:

Neste ponto, o conversor A/D possui 34 sinais, que são:

- 1) A versão analógica do bit MSB;
- 2) A versão analógica do bit 2SB;
- 3) A versão analógica do bit 3SB;
- 4) O sinal auxiliar S90;
- 5) Os 30 sinais auxiliares obtidos por interpolação (veja a Fig. 1.12).

Cada um destes sinais passará por um “latch” mestre-escravo e se tornará um sinal digital. Um circuito básico utilizado para um “latch” (mestre ou escravo) na tecnologia bipolar pode ser visto na Fig. 1.13. O princípio de operação do circuito é o seguinte: O sinal diferencial de entrada é amplificado pelo circuito pré-amplificador composto pelo par diferencial  $T_1$  e  $T_2$  e aparece nos resistores de coletor  $R_1$  e  $R_2$ . O outro par diferencial composto por  $T_3$  e  $T_4$  retém e transforma o sinal diferencial de entrada em um sinal digital de saída. O par diferencial  $T_5$  e  $T_6$  comuta o circuito do modo de amplificação para o modo de conversão. Quando o sinal do “clock” está em nível alto (“1”), a corrente passa através de  $T_5$  e o circuito pré-amplificador é ativado. No momento da conversão, o sinal do “clock” vai para nível baixo (“0”) e a corrente passa através do transistor  $T_6$ . O par diferencial composto por  $T_3$  e  $T_4$  fica ativo. A realimentação positiva, neste sistema, resulta em alto ganho e sensibilidade do circuito, gerando um sinal digital na saída. A otimização do circuito permite a utilização de altas frequências no sinal do “clock”. O comportamento do “latch” irá influenciar a razão de erro de bit (“bit error rate” – BER) do conversor A/D [08].

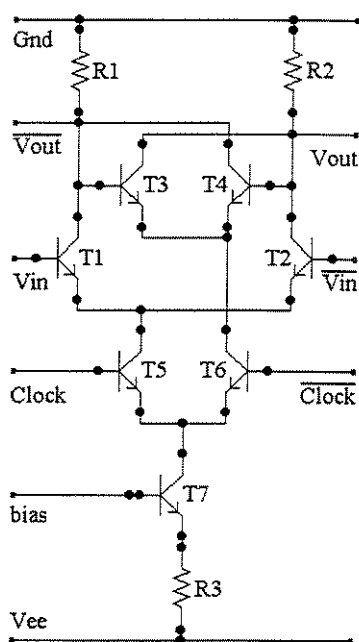


Figura 1.13 – “Latch” Básico

Nos conversores A/D, os “latches” realizam muitas decisões durante o processo de conversão. Isto é especialmente crítico nos conversores A/D de alta taxa de conversão. No momento em que algum “latch” toma uma decisão errada, o código interno também fica errado e é convertido para um correspondente código errado de saída. Algumas vezes, o “latch” também pode assumir a condição de meta-estado [08]. O meta-estado é um fenômeno tipicamente associado com sistemas digitais binários, particularmente aqueles que utilizam “latch”. Em um “latch”

é esperado que haja dois estados distintos, que representam o estado lógico “1” e um estado lógico “0”. A maioria dos “latches” também é capaz de gerar um terceiro estado, com nível lógico indeterminado entre os níveis lógicos “1” e “0”. Isto pode ocorrer quando o tempo de decisão (“setup” e “hold”) do “latch” não é respeitado e a borda do sinal do “clock” ocorre quando o sinal de entrada ainda está em transição. O estado indeterminado é descrito como sendo um meta-estado, porque a condição irá eventualmente mudar, não necessariamente de uma maneira monotônica, para um dos estados lógicos válidos [17]. Desta forma, a ocorrência de um estado lógico errado ou inválido (meta-estado) na saída de algum “latch” causará erro no código interno do conversor A/D que, se não for corrigido, será propagado ao código de saída.

Para análise da informação sobre o erro nos “latches” é utilizado o BER, que define o número de erros de decisão ocorridos no conversor A/D em função da sua frequência de amostragem. Por exemplo, um conversor A/D de alta qualidade deve ter o BER entre  $10^{-10}$  e  $10^{-15}$  erros/amostragem do “latch” [08]. O projeto dos “latches” irá influenciar o valor do BER e a

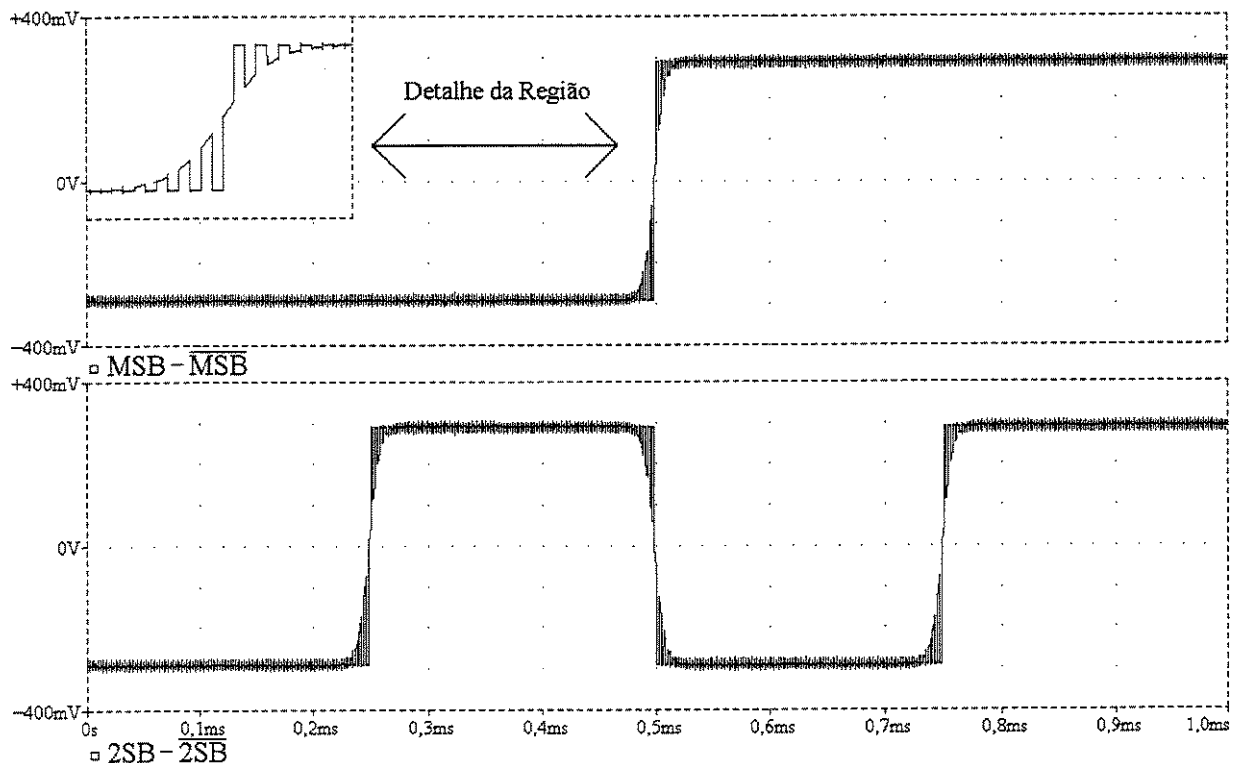


Figura 1.14 – MSB e 2SB

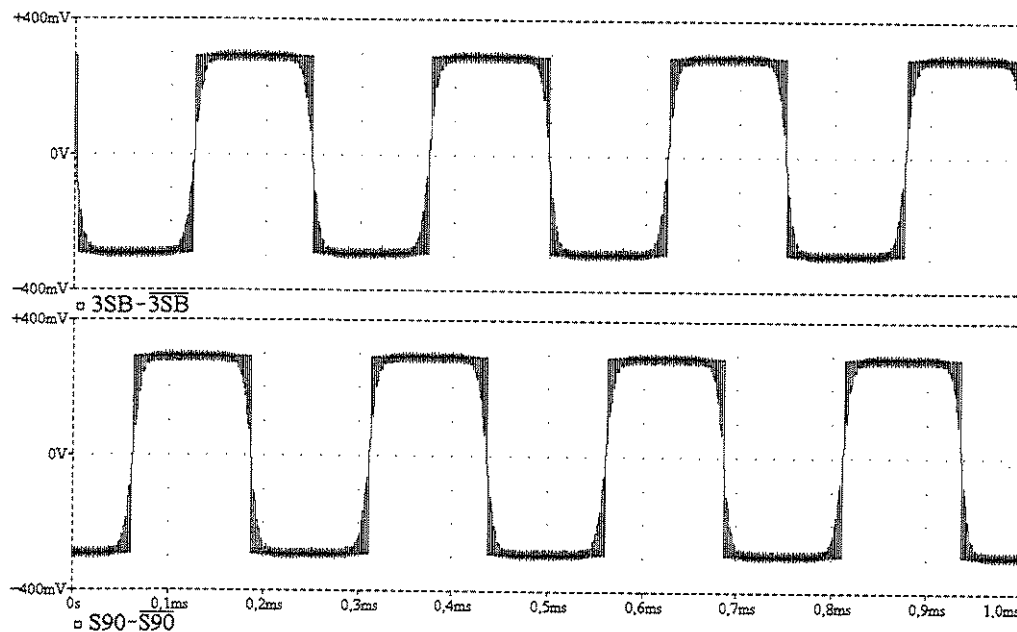


Figura 1.15 –  $3SB$  e  $S90$

máxima frequência de amostragem do conversor A/D.

Na saída dos “latches” mestres, os sinais do ( $MSB-\overline{MSB}$ ), ( $2SB-\overline{2SB}$ ) (veja a Fig. 1.14), ( $3SB-\overline{3SB}$ ) e ( $S90-\overline{S90}$ ) (veja a Fig. 1.15) e os sinais interpolados (veja a Fig. 1.16) já estão digitalizados e têm o mesmo nível e frequência do ( $3SB-\overline{3SB}$ ). Os sinais ( $3SB-\overline{3SB}$ ) e ( $S90-\overline{S90}$ ) junto com os sinais interpolados formam um código circular (veja a Fig. 1.16 e a Tabela 1.2) [06 e 08], também chamado de código termômetro cíclico [11 e 16].

Todas estas figuras apresentam um “ruído” referente ao sinal de “clock”. A Fig. 1.14 também mostra um detalhe interessante no sinal do ( $MSB-\overline{MSB}$ ), no qual é bastante perceptível os instantes em que o “latch” faz o “sample” e o “hold”. Isto ocorrerá frequentemente nas figuras com estes sinais. Após passar pelos “latches” escravos (eles têm o sinal de “clock” invertido em relação ao sinal de “clock” utilizado nos “latches” mestres), este “ruído” fica muito minimizado.

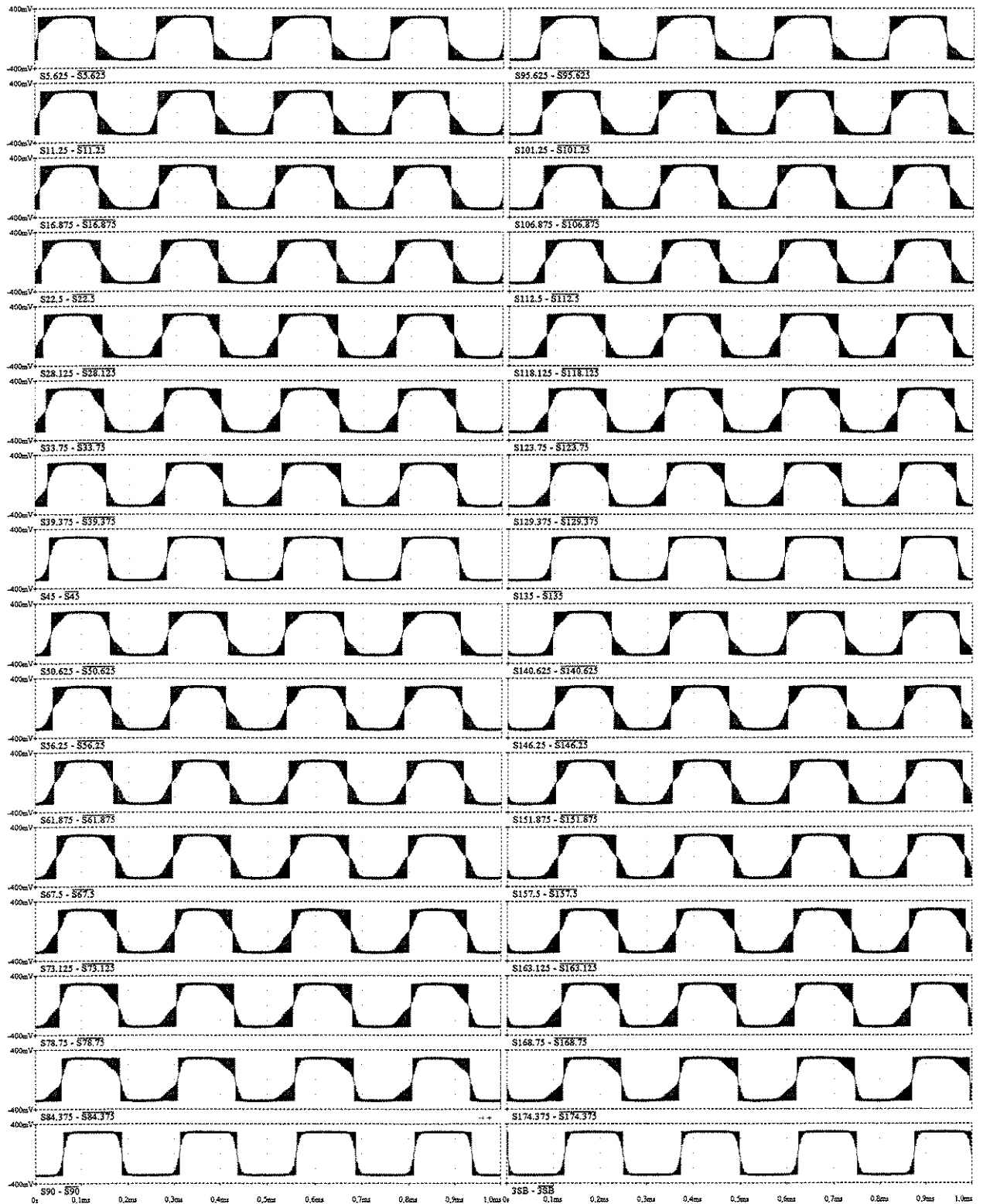


Figura 1.16 – Sinais 3SB, S90 e Interpolados

Tabela 1.2 – *Código Circular*[illegible]

## 1.6 – Erro de Código:

O código circular utilizado nos conversores A/D tipo "folding" com interpolação é semelhante ao código termômetro utilizado nos conversores A/D tipo "flash". Uma comparação simplificada entre os dois códigos pode ser vista na Tabela 1.3. Ambos os códigos estão sujeitos a terem erros, porque problemas diretamente relacionados com a razão de conversão e/ou frequência do sinal analógico de entrada podem causar erro de decisão nos “latches” comparadores (do conversor A/D tipo “flash”) e nos “latches” mestre-escravo (do conversor A/D tipo “folding”) e corromper o código circular interno do conversor A/D. Isto resultará em um correspondente código errado na saída. Erros deste tipo são chamados de erros de “bolha” ("bubble"), porque eles se assemelham a bolhas no “mercúrio” do código termômetro [18]. Os erros de bolha se propagam adiante para o codificador e isto resulta em nível lógico inválido na saída. A Tabela 1.4 mostra os 4 padrões de erros de bolha mais frequentes para ambos os códigos. A maior probabilidade de ocorrência de padrão de erro de bolha é a do erro 1, seguido pelo erro 2 e erro 3 da Tabela 1.4 [22]. Nos conversores A/D tipo “flash”, o erro 4 representa uma séria degradação e não deve ocorrer dentro do mesmo bloco de “latches” comparadores [18]. Em alguns conversores A/D tipo “flash”, os “latches” comparadores são arranjados em alguns blocos para conveniência de “layout” e codificação. Nos conversores A/D tipo duplo “folding” com

Tabela 1.3 – *Exemplos dos Códigos*

|   | Código Termômetro | Código Circular |
|---|-------------------|-----------------|
| 0 | 0 0 0 0 0 0 0     | 0 0 0 0         |
| 1 | 0 0 0 0 0 0 1     | 0 0 0 1         |
| 2 | 0 0 0 0 0 1 1     | 0 0 1 1         |
| 3 | 0 0 0 0 1 1 1     | 0 1 1 1         |
| 4 | 0 0 0 1 1 1 1     | 1 1 1 1         |
| 5 | 0 0 1 1 1 1 1     | 1 1 1 0         |
| 6 | 0 1 1 1 1 1 1     | 1 1 0 0         |
| 7 | 1 1 1 1 1 1 1     | 1 0 0 0         |

Tabela 1.4 – *Exemplos de Erros de Bolha*

|         |   |   |   |   |   |   |   |           |
|---------|---|---|---|---|---|---|---|-----------|
| Correto | 0 | 1 | 1 | 1 | 1 | 1 | 1 | sem bolha |
| Erro 1  | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 bolha   |
| Erro 2  | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 bolha   |
| Erro 3  | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 2 bolhas  |
| Erro 4  | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 2 bolhas  |

interpolação, o conjunto de “latches” mestre-escravo que transforma a informação analógica interpolada em dados digitais de saída forma um único bloco.

Na decodificação do código circular é detectada a transição de zero para um e os circuitos irão endereçar uma linha da estrutura codificadora binária tipo “ROM”. Se existir algum erro de bolha, haverá pelo menos dois pontos de transição de zero para um no código e os circuitos irão endereçar duas linhas da estrutura codificadora binária tipo “ROM”. Endereçamento simultâneo de duas linhas da estrutura codificadora binária tipo “ROM” causará uma descontinuidade, ou “glitch”, na forma de onda reconstruída, porque ocorrerá um nível lógico inválido.

Considerando que os conversores A/D do tipo “flash” são mais utilizados, a maioria dos trabalhos publicados sobre correção do erro de bolha se referem ao código termômetro. Existem vários métodos para corrigir o erro de bolha no código termômetro. O método mais comum de suprimir erros de bolha é usar uma porta com três entradas para endereçar a estrutura codificadora binária tipo “ROM” [18 e 19]. Algumas vezes é usada a codificação Gray porque, devido a sua natureza simétrica de codificação, a diferença entre o código correto e o código incorreto é pequena quando ocorre o erro de bolha [20]. A maior razão para que a codificação Gray não seja usada exclusivamente é que ela requer uma lógica adicional para ser convertida de volta para o código binário. Recentemente, as propostas para correção do erro de bolha ficaram mais sofisticadas e alguns exemplos de correção de erro de bolha foram apresentados em [18, 20, 21 e 22]. A maioria das propostas irá remover com sucesso os erros de bolha 1, 2 e 3 da Tabela 1.4, mas algumas irão falhar no caso extremo do erro 4.

Normalmente, o esquema de correção de erro pode ser pensado como um processo de votação. Cada saída de “latch” é examinada em relação às duas saídas vizinhas próximas, e a saída é trocada se ela discordar de ambas. Porém, alguns destes circuitos requerem um grande número de elementos. As propostas de correção dos erros de bolha no código termômetro devem ser adaptadas para serem usadas no código circular porque os códigos são diferentes.

Para aplicação específica no código circular, existe uma correção analógica para proteger um código circular em um conversor A/D duplo “folding” com interpolação [07 e 08]. Esta técnica é baseada no cálculo da média de um conjunto de três saídas dos “latches” mestre antes de alimentar a informação nos “latches” escravo. Nesta proposta, são necessários somente quatro transistores por “latch” escravo e o consumo não é incrementado no total porque esta técnica faz uso da corrente existente no “latch” escravo (veja a Fig. 1.17). Esta proposta irá remover com sucesso os erros de bolha 1, 2 e 3 da Tabela 1.3, porém, irá falhar no caso extremo do erro 4, mas

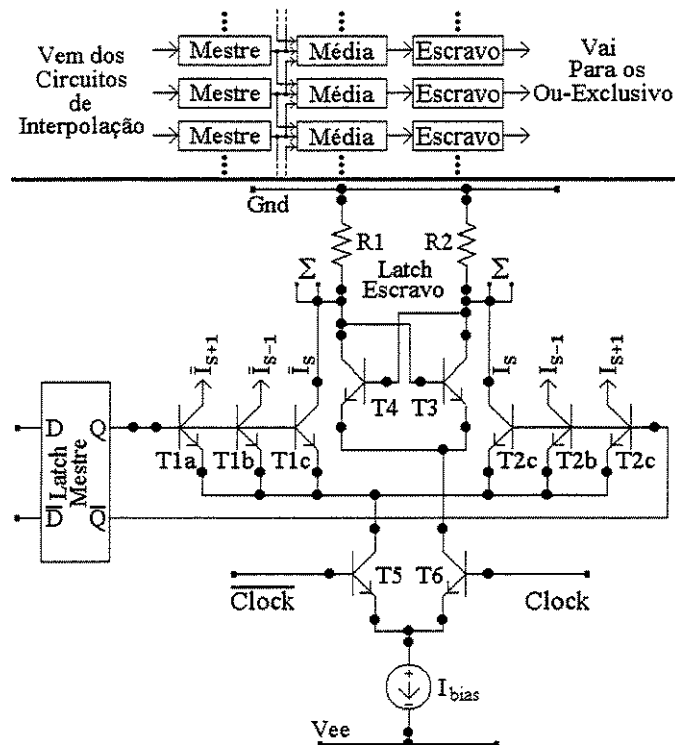


Figura 1.17 – Correção Analógica de Erro [07 e 08]

corrigirá um eventual meta-estado no “latch” mestre [08].

### 1.7 – Circuito Codificador:

O código circular pode ser convertido em uma forma binária usando portas “ou-exclusivo” (EXOR) (Fig. 1.18) e uma estrutura codificadora binária tipo “ROM”. As saídas deste codificador tipo “ROM” são os bits menos significativos. Na operação “ou-exclusivo”, realizada entre as saídas adjacentes dos “latches” (Tabela 1.5), são gerados 32 sinais que podem ser vistos na Fig. 1.19. Nesta figura, é mostrado o sinal do  $(3SB-3SB)$  para servir como referência aos

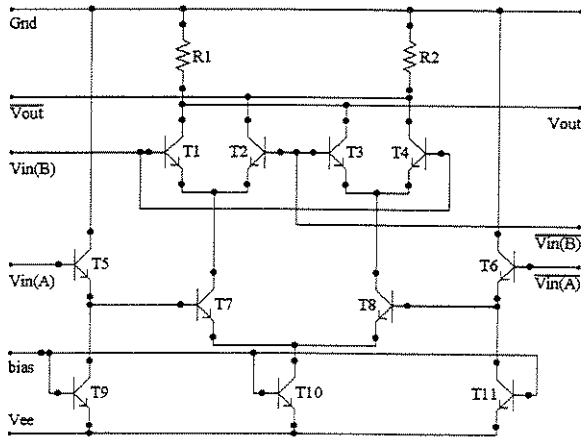


Figura 1.18 – Porta Ou-Exclusivo

sinais de saída das portas “ou-exclusivo”. Estes sinais também foram divididos em dois gráficos para melhor visualização. É possível verificar que os 32 sinais se repetem num período de meio 3SB, o que corresponde a um 4SB. A correta composição destes sinais irá formar os 5 bits menos significativos.

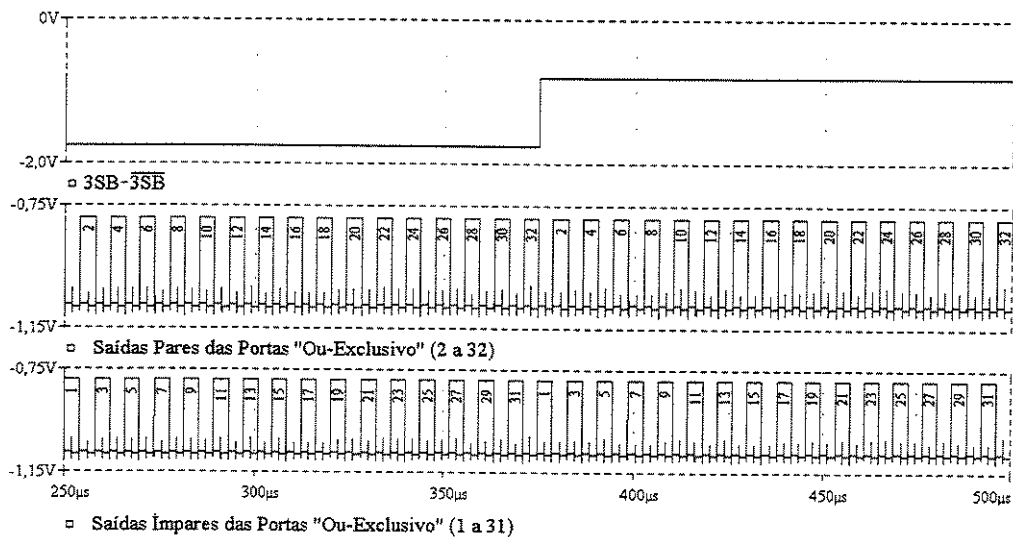


Figura 1.19 – Sinais na Saída das Portas Ou-Exclusivo

Tabela 1.5 – Saídas das Portas Ou-Exclusivo

[illegible]

A estrutura tipo “ROM” combina os sinais de saída das portas “ou-exclusivo” em 10 sinais complementares (veja a Fig. 1.20) [06 e 08]. Estes sinais são: 4SB,  $\overline{4SB}$ , 5SB,  $\overline{5SB}$ , 6SB,  $\overline{6SB}$ , 7SB,  $\overline{7SB}$ , LSB e  $\overline{LSB}$ . Veja, a seguir, as combinações dos sinais de saídas das portas “ou-exclusivo” na estrutura tipo “ROM” e compare a Tabela 1.5 com as Fig. 1.20 e 1.21:

$$4SB \Rightarrow (32 + 31 + 30 + 29 + 28 + 27 + 26 + 25 + 24 + 23 + 22 + 21 + 20 + 19 + 18 + 17)$$

$$\overline{4SB} \Rightarrow (16 + 15 + 14 + 13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)$$

$$5SB \Rightarrow (32 + 31 + 30 + 29 + 28 + 27 + 26 + 25 + 16 + 15 + 14 + 13 + 12 + 11 + 10 + 9)$$

$$\overline{5SB} \Rightarrow (24 + 23 + 22 + 21 + 20 + 19 + 18 + 17 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)$$

$$6SB \Rightarrow (32 + 31 + 30 + 29 + 24 + 23 + 22 + 21 + 16 + 15 + 14 + 13 + 8 + 7 + 6 + 5)$$

$$\overline{6SB} \Rightarrow (28 + 27 + 26 + 25 + 20 + 19 + 18 + 17 + 12 + 11 + 10 + 9 + 4 + 3 + 2 + 1)$$

$$7SB \Rightarrow (32 + 31 + 28 + 27 + 24 + 23 + 20 + 19 + 16 + 15 + 12 + 11 + 8 + 7 + 4 + 3)$$

$$\overline{7SB} \Rightarrow (30 + 29 + 26 + 25 + 22 + 21 + 18 + 17 + 14 + 13 + 10 + 9 + 6 + 5 + 2 + 1)$$

$$LSB \Rightarrow (32 + 30 + 28 + 26 + 24 + 22 + 20 + 18 + 16 + 14 + 12 + 10 + 8 + 6 + 4 + 2)$$

$$\overline{LSB} \Rightarrow (31 + 29 + 27 + 25 + 23 + 21 + 19 + 17 + 15 + 13 + 11 + 9 + 7 + 5 + 3 + 1)$$

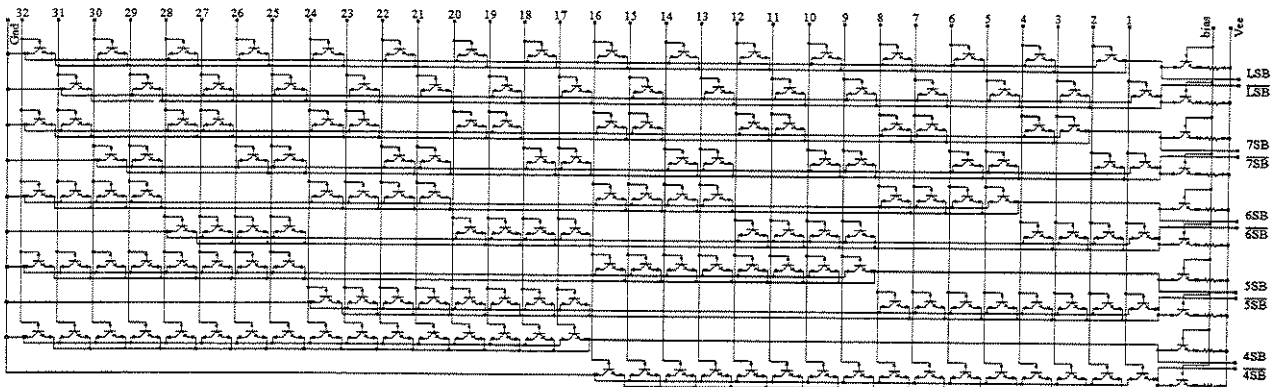


Figura 1.20 – Estrutura Tipo ROM

## 1.8 – Sincronismo de Bits:

A arquitetura “folding” pode ser considerada como uma arquitetura “two-step” porque a conversão do sinal analógico é dividida em duas etapas. Isto é, a conversão do sinal analógico consiste em um estágio de conversão para os bits mais significativos (“coarse bits”) e

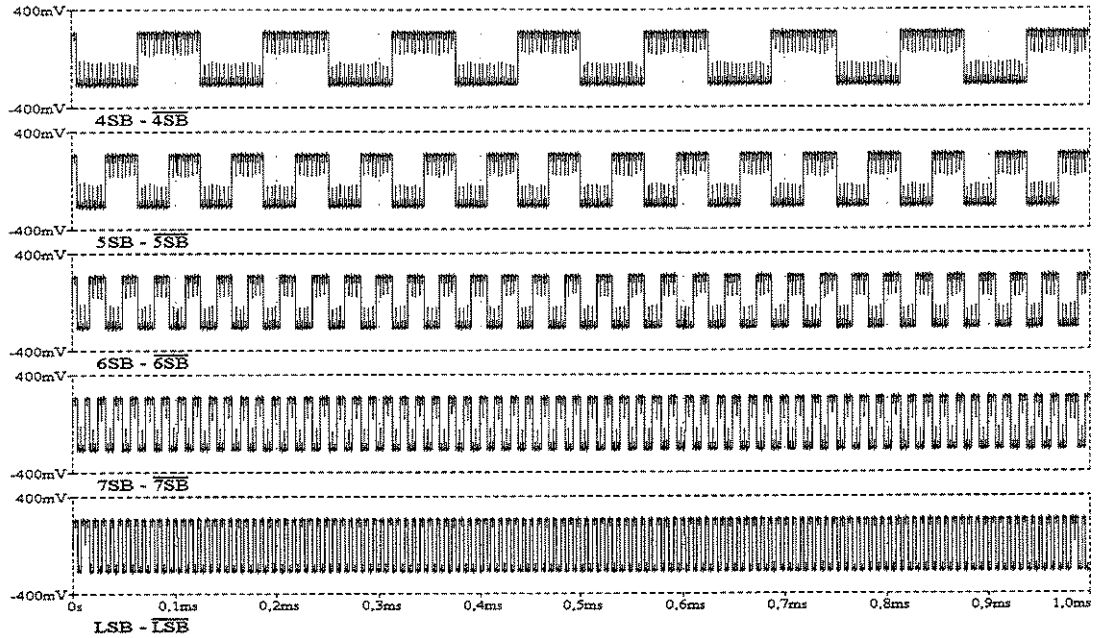


Figura 1.21 – Sinais de Saída da Estrutura Tipo ROM

outro estágio de conversão para os bits menos significativos (“fine bits”), mas as duas etapas de conversão são realizadas em paralelo [12].

Um pequeno “offset” de tensão ou atraso entre os “latches” dos bits mais significativos e dos bits menos significativos pode gerar um código errado na saída do conversor A/D e causar “glitch” no sinal reconstruído porque os “latches” não mudam de estado exatamente no mesmo instante (veja a Fig. 1.22). Esta fonte de erro necessita de um sistema específico de

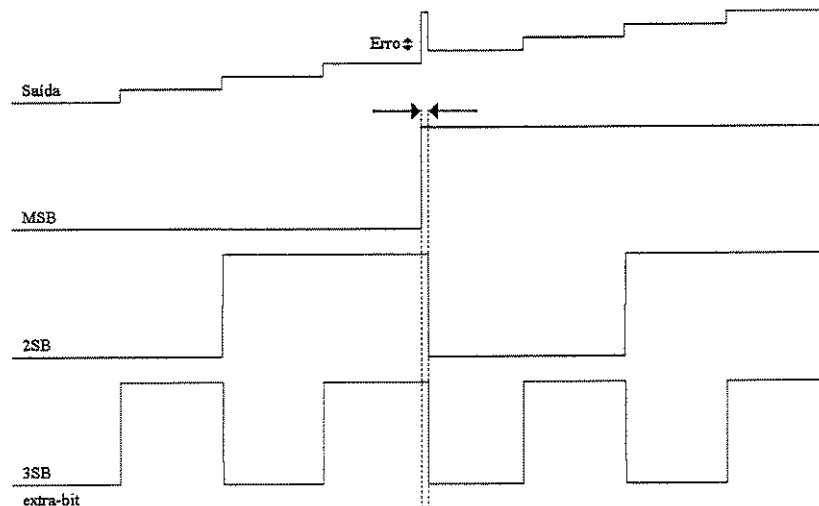


Figura 1.22 – Desalinhamento dos “Latches”

Tabela 1.6 – Correção de Erro

| Bit/Região | “A”              | “B”              | “C” |
|------------|------------------|------------------|-----|
| MSB        | $\overline{3SB}$ | -                | -   |
| 2SB        | 3SB              | $\overline{3SB}$ | 3SB |

correção para sincronismo dos bits. Este erro se apresenta de forma acentuada nos instantes de transição dos bits mais significativos por causa do maior peso relativo na reconstrução do sinal.

A solução é detectar estes instantes de transição e, no momento adequado, substituir os bits mais significativos por um valor apropriado (veja a Fig. 1.23). Neste caso, utiliza-se o terceiro bit como referência porque ele também participa da geração dos bits menos significativos através da interpolação. Desta maneira, a informação oriunda do terceiro bit é usada para corrigir qualquer erro de temporização nos bits mais significativos antes deles serem aplicados nos “buffers” de saída “ECL”. Se os bits mais significativos estão na região de transição, eles são substituídos por um valor apropriado baseado no terceiro bit. Isto é, os bits mais significativos são substituídos pelo terceiro bit ou pelo seu complemento dependendo da região de transição. Neste caso, o terceiro bit é chamado de “extra-bit” e os bits mais

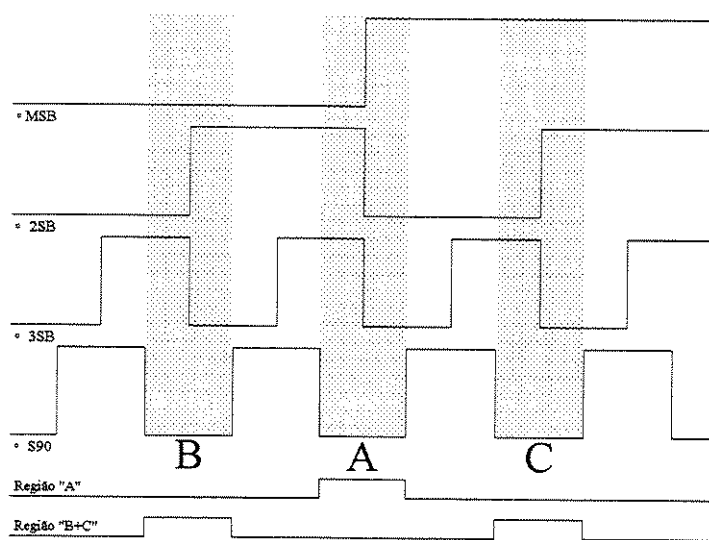


Figura 1.23 – Regiões de Transição

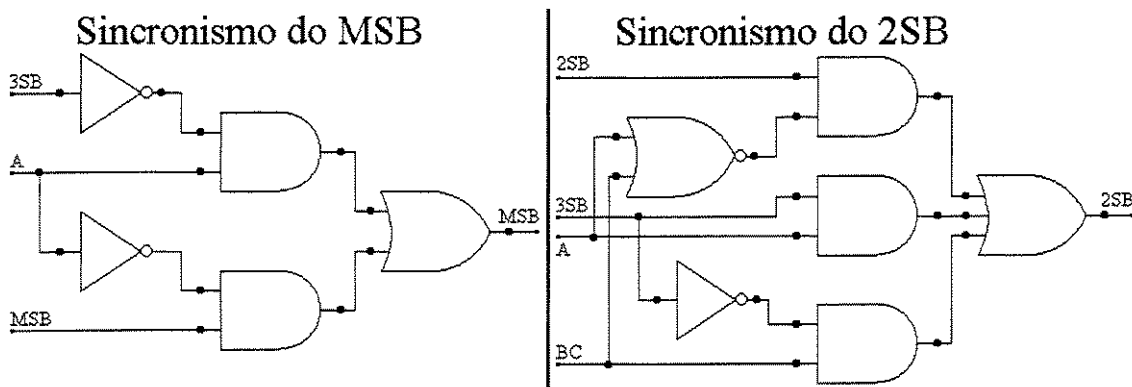


Figura 1.24 – Circuitos Lógicos de Sincronismo

significativos são sincronizados com o resto do sistema usando o “extra-bit” (veja a Fig. 1.22).

Na Fig. 1.23 é possível ver as três regiões de transição que podem ser facilmente identificadas. Na região “A”, no instante de transição, o sinal do MSB deve ser substituído pelo  $\overline{3SB}$  e o 2SB deve ser substituído pelo 3SB. Nas regiões “B” e “C” não existe transição do MSB e o 2SB deve ser substituído pelo  $\overline{3SB}$ . Todas combinações possíveis podem ser vistas na Tabela 1.6 e a Fig. 1.24 mostra os circuitos lógicos a serem utilizados nas substituições dos bits. Apesar das regiões de transição serem facilmente identificáveis, é necessário um circuito extra para a sua exata e correta identificação (se a região é “A” ou “B” ou “C”), de maneira que possa ser utilizada nos circuitos lógicos de correção.

### 1.9 – Portas de Saída:

Os bits mais significativos, após receberem a correção de sincronismo, vão para as portas de saída. Os bits menos significativos da saída da estrutura decodificadora tipo “ROM” também vão para as portas de saída. A porta de saída (veja a Fig. 1.25) tem por objetivo servir de “buffer” e ajustar aos níveis “ECL” o sinal de saída em código binário. A Fig. 1.26 mostra os sinais de saída resultantes da simulação de um conversor A/D de 8 bits tipo “folding” com interpolação. Apesar de existirem algumas variações na topologia deste tipo de conversor, de forma geral, é seguida esta configuração. No próximo capítulo serão mostrados os tipos de interpolação.

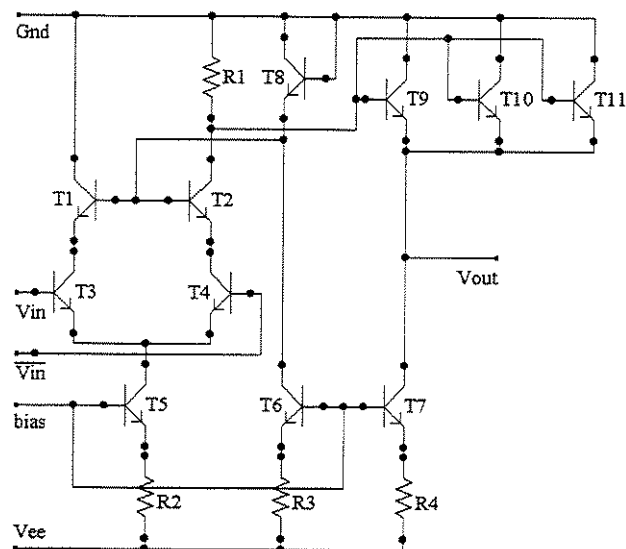


Figura 1.25 – *Porta de Saída*

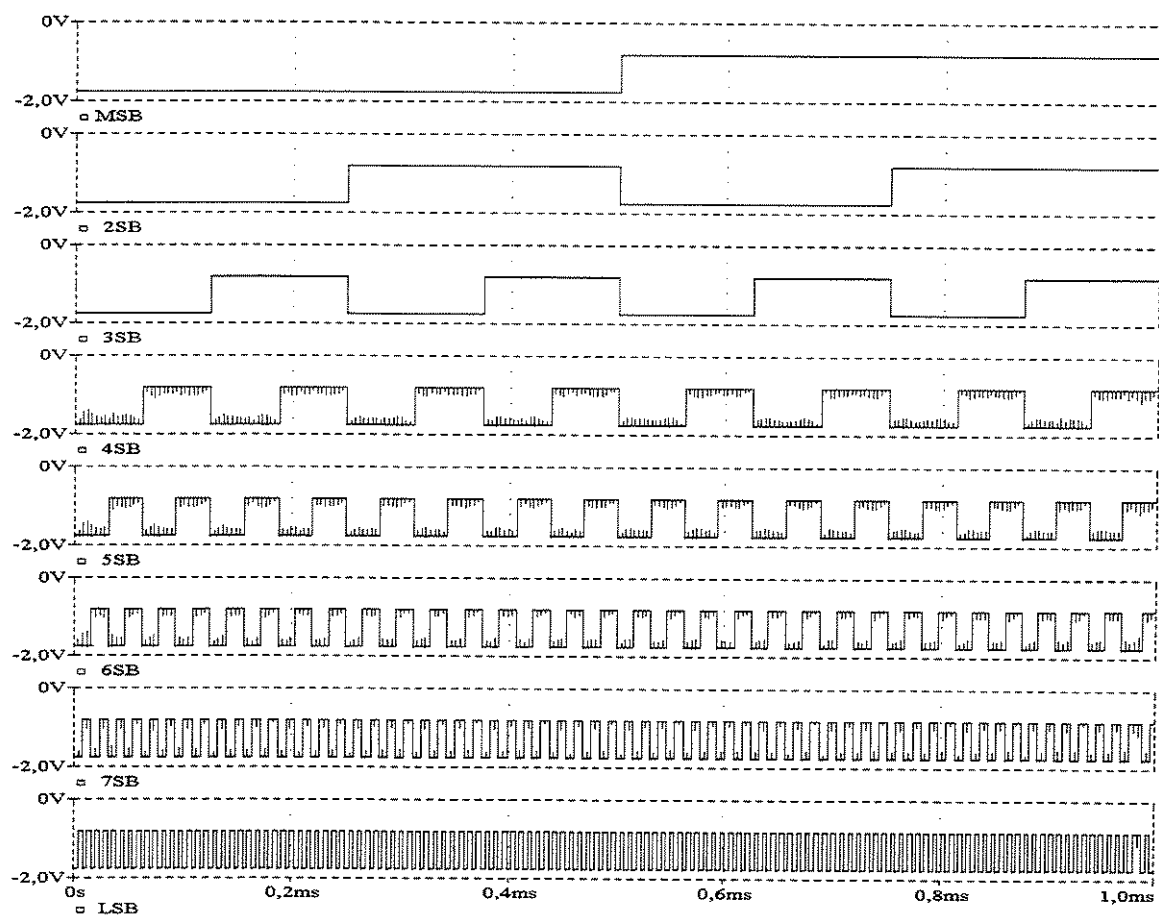


Figura 1.26 – *Sinais de Saída*

## Capítulo 2

### Interpolação

#### 2.1 – Características Gerais:

Na arquitetura de duplo “folding” com interpolação, os bits mais significativos são determinados pela quantização do sinal de entrada usando um circuito “folding”, enquanto os bits menos significativos são obtidos pela técnica de interpolação. A interpolação reduz a complexidade dos circuitos sem incrementar a razão de “folding” do sistema. A pequena distorção sofrida pelo sinal interpolado não é importante porque, se o cruzamento de zero está correto, os “latches” irão decidir corretamente [05]. Isto ocorre porque as partes superior e inferior do sinal interpolado não ficam completamente corretas (veja a Fig. 2.1) [09]. A maioria das soluções empregadas para implementar a técnica de interpolação utiliza a interpolação resistiva. Alternativamente, existe uma técnica de interpolação por divisão de corrente (com transistores NMOS) [11]; e esta nova proposta propõe fazer a interpolação nos próprios “latches”

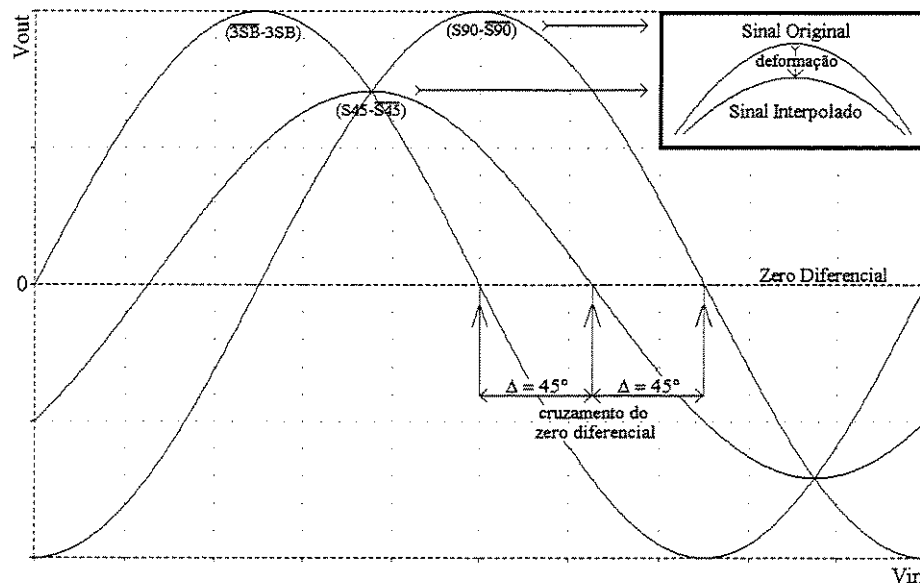


Figura 2.1 – Distorção na Interpolação

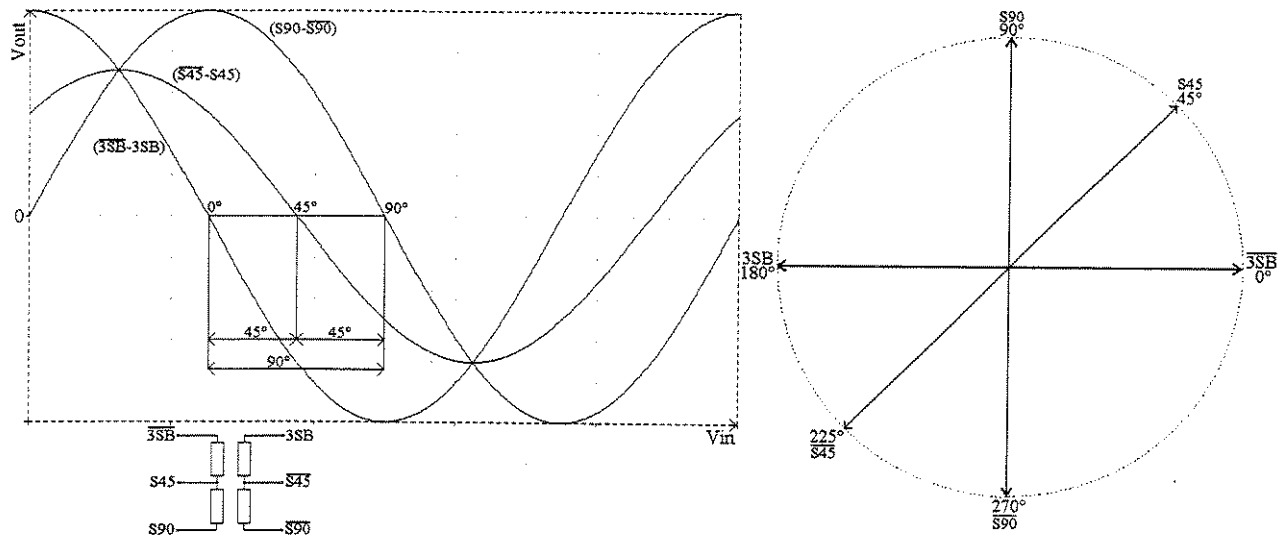


Figura 2.2 – Interpolação Básica

mestres do conversor A/D. Para melhor entendimento da nova técnica de interpolação, é apresentada a sua versão inicial e a sua versão final que é aprimorada em relação à primeira versão. A vantagem é bem evidente, desde que a nova técnica proposta de interpolação foi desenvolvida para evitar a rede de interpolação resistiva e seus problemas com a mesma eficiência e um consumo semelhante de potência.

## 2.2 – Interpolação Resistiva:

A interpolação pode ser facilmente implementada pela inserção de uma rede resistiva de interpolação entre dois sinais de saída do “folding encoder”. Um exemplo de interpolação resistiva é mostrado na Fig. 2.2. Dois sinais, com diferença de fase de 90°, podem produzir outros sinais com diferença de fase intermediárias. Na Fig. 2.2, aparecem dois conjuntos resistivos de interpolação que fazem a interpolação entre (3SB com S90) e (3SB com S90), obtendo os sinais S45 e S45. Para aumentar a resolução do conversor em 1 bit, seria necessário mais os sinais S135 e S135 (veja a Fig. 2.3), que podem ser obtidos pela interpolação entre (3SB com S90) e (3SB com S90). O número de resistores usados em um conjunto resistivo de interpolação é chamado de

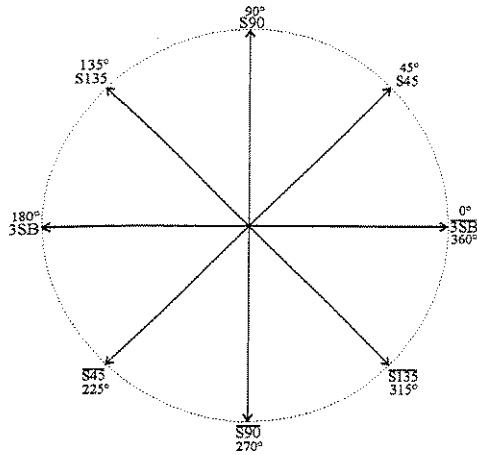


Figura 2.3 – Diagrama de Fase 1

razão de interpolação. No exemplo anterior, existem dois resistores em cada conjunto resistivo de interpolação e, desta forma, a razão de interpolação é igual a dois. A resolução do conversor A/D é aumentada em  $\log_2(\text{razão de interpolação})$  [07] e, para o exemplo dado, o valor é  $\log_2(2) = 1$ . Portanto, a resolução é aumentada em 1 bit, utilizando-se quatro conjuntos de interpolação resistiva com dois resistores

em cada uma delas. Este aumento de resolução seria conseguido com a combinação lógica dos sinais interpolados.

Como já foi apresentado no capítulo anterior, a implementação prática da interpolação é feita com sinais diferenciais. Isto torna possível a análise dos sinais pela diferença entre duas

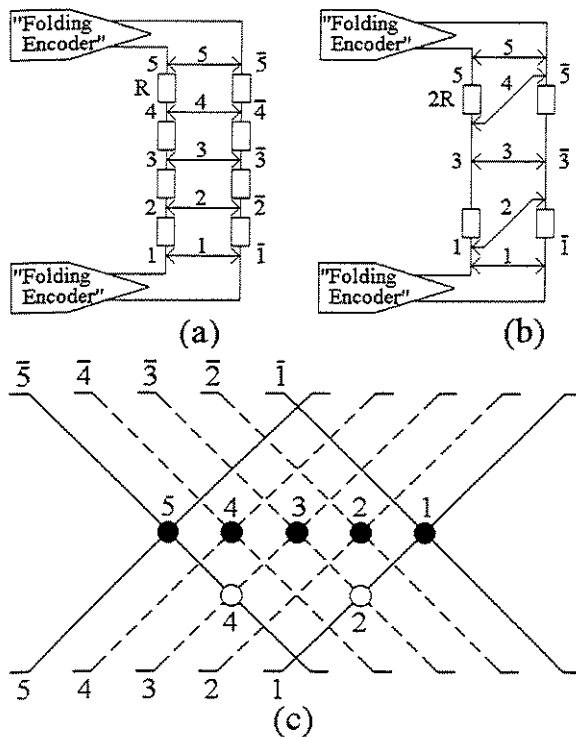


Figura 2.4 – Implementação da Interpolação

voltagens e não pelo seu valor fixo de referência de voltagem [05]. Na Fig. 2.4 é possível ver dois circuitos de interpolação. Em ambos, é possível recuperar a mesma informação através da interpolação. No circuito “(a)”, com oito resistores, há dois sinais diferenciais gerados (1/1̄ e 5/5̄) e são obtidos outros três sinais diferenciais através da interpolação (2/2̄, 3/3̄ e 4/4̄). No circuito “(b)”, com quatro resistores, há dois sinais diferenciais gerados (1/1̄ e 5/5̄)

e são obtidos os mesmos três sinais diferenciais através da interpolação ( $2/\overline{2}$ ,  $3/\overline{3}$  e  $4/\overline{4}$ ). Isto ocorre porque, na prática, não é necessário ter fisicamente todos os três sinais diferenciais interpolados. A mesma informação (do cruzamento por zero) também pode ser encontrada usando-se os dois sinais diferenciais gerados ( $1/\overline{1}$  e  $5/\overline{5}$ ) e somente um sinal diferencial interpolado ( $3/\overline{3}$ ). Na Fig. 2.4(c), é possível verificar que o cruzamento por zero dos sinais diferenciais interpolados ( $2/\overline{2}$  e  $4/\overline{4}$ ) é redundantes e a mesma informação pode ser obtida entre os sinais ( $1$  com  $\overline{3}$ ) e ( $3$  com  $\overline{5}$ ), respectivamente. A vantagem é a diminuição do circuito e a consequente compactação do “layout”. Esta compactação é especialmente benéfica porque os sinais analógicos, neste ponto do conversor A/D tipo “folding” com interpolação, têm alta frequência (um múltiplo da frequência do sinal analógico de entrada, conforme razão de “folding”) e a capacitância parasita do circuito resistivo de interpolação deve ser minimizada [05].

A Fig. 2.5 mostra o exemplo de uma rede resistiva de interpolação. As saídas de estágios seguidores de emissor, servem de “buffer” para a rede de interpolação resistiva. A impedância de saída desta rede, que alimenta o sinal na entrada dos “latches”, varia de “0” a “R” (onde “R” é o valor do resistor de interpolação). A carga capacitiva de entrada dos “latches” na rede de interpolação resulta em um “delay” variável no sinal que pode, facilmente, atingir valores não

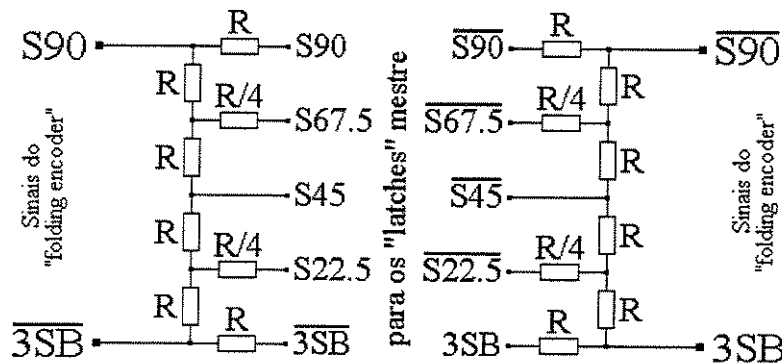


Figura 2.5 – Rede Resistiva de Interpolação

permitidos pelo sistema. Portanto, são colocados resistores adicionais de valor igual a “R” e “R/4” em série com as saídas (da rede resistiva) para fazer todas as impedâncias iguais a “R” (veja a Fig. 2.5 e a Tabela 2.1) [06 e 08]. Desta maneira, todos os “latches” terão um mesmo “delay”.

Tabela 2.1 – Impedâncias de Saída

| Saídas                     | Impedância          |
|----------------------------|---------------------|
| 3SB e $\overline{3SB}$     | R                   |
| S22.5 e $\overline{S22.5}$ | $R/4 + R // 3R = R$ |
| S45 e $\overline{S45}$     | $2R // 2R = R$      |
| S67.5 e $\overline{S67.5}$ | $R/4 + R // 3R = R$ |
| S90 e $\overline{S90}$     | R                   |

Um exemplo prático de uma rede resistiva de interpolação pode ser visto na Fig. 2.6. Nesta figura, dois sinais diferenciais são gerados pelo “folding encoder” (3SB/ $\overline{3SB}$  e S90/ $\overline{S90}$ ) e

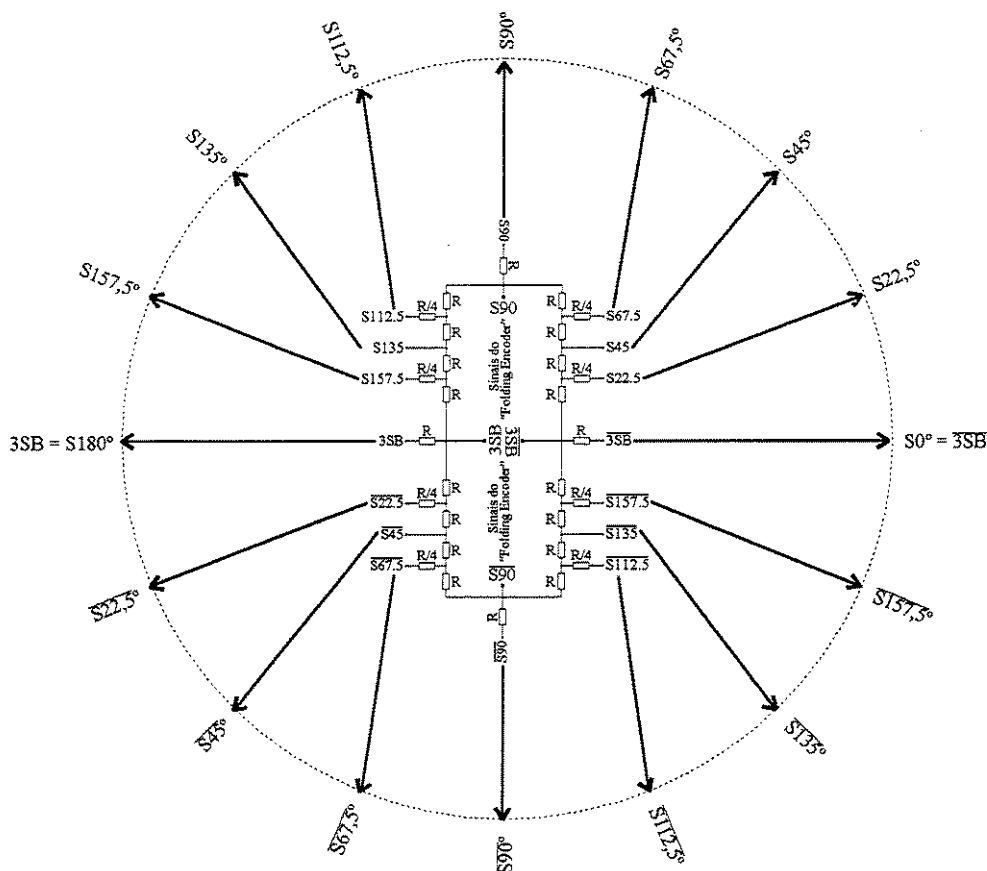


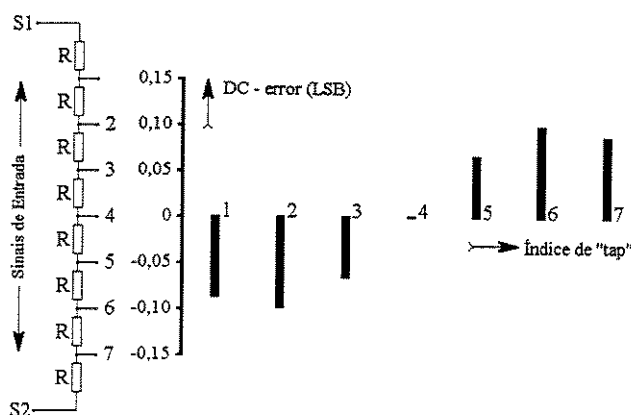
Figura 2.6 – Sinais na Rede Resistiva de Interpolação

utilizados para se obterem outros seis sinais diferenciais interpolados ( $S_{22.5}/\overline{S_{22.5}}$ ,  $S_{45}/\overline{S_{45}}$ ,  $S_{67.5}/\overline{S_{67.5}}$ ,  $S_{112.5}/\overline{S_{112.5}}$ ,  $S_{135}/\overline{S_{135}}$  e  $S_{157.5}/\overline{S_{157.5}}$ ). Estes sinais, mais os sinais dos bits mais significativos tornariam possível a implementação de um conversor A/D tipo “folding” com interpolação de 6 bits.

A técnica de interpolação resistiva é um método eficiente, em termos de consumo de potência, para gerar os sinais interpolados porque, sendo um circuito diferencial, as correntes (dos sinais) que passam através dos resistores são forçadas a fluir em círculo e nenhuma corrente é desperdiçada [09].

Por causa da não linearidade na curva de transferência “dc” de um CDP (“coupled differential pair”) no amplificador “folding”, existe um pequeno erro “dc” nos sinais definidos pela interpolação resistiva [07]. O erro, que não deve exceder  $\pm 0,1$  LSB, é aceitável para a maioria das aplicações. Isto é, o erro que é introduzido devido a precisão limitada dos sinais nos circuitos “folding” deve ser inferior a  $\pm 0,1$  LSB para um conversor A/D de 8 bits [08]. A Fig. 2.7 mostra este erro expresso em LSB.

Uma desvantagem da interpolação resistiva é que os valores dos resistores devem ser pequenos nos conversores A/D muito rápidos, o que dificulta o projeto de alimentação do circuito de interpolação resistiva [11]. Outro aspecto a ser considerado no projeto é o “delay” na



propagação do sinal e as capacitâncias parasitas. A vantagem da interpolação resistiva é a sua eficiência no consumo de potência [12].

Figura 2.7 – Erro “dc” na Rede Resistiva de Interpolação

### 2.3 – Interpolação em Modo de Corrente:

A técnica de interpolação em modo de corrente foi utilizada em conversores A/D com tecnologia CMOS. Esta forma de interpolação utiliza a técnica de divisão de corrente para fazer a interpolação entre dois sinais (veja a Fig. 2.8). As correntes de saída de dois sinais ( $I_A$  e  $I_B$ ) são divididas por quatro, utilizando-se quatro transistores NMOS idênticos ligados em configuração “fonte” comum [11]. A corrente  $I_A/4$  é somada à corrente  $I_B/4$  para formar o sinal interpolado. O sinal original é constituído por dois quartos da corrente de entrada (por exemplo:  $2 \cdot I_A/4 = I_A/2$ ). No exemplo da Fig. 2.8, a ordem de interpolação ( $n$ ) é dois. Ordens mais altas de interpolação podem ser implementadas utilizando-se transistores divisores de corrente adicionais. Uma desvantagem desta técnica é que são necessários  $2^n$  transistores, comparando com “ $n$ ” resistores da técnica convencional de interpolação através do uso de uma rede resistiva [11]. Na Fig. 2.9, é mostrado o exemplo de implementação parcial de uma interpolação de segunda ordem. As fontes

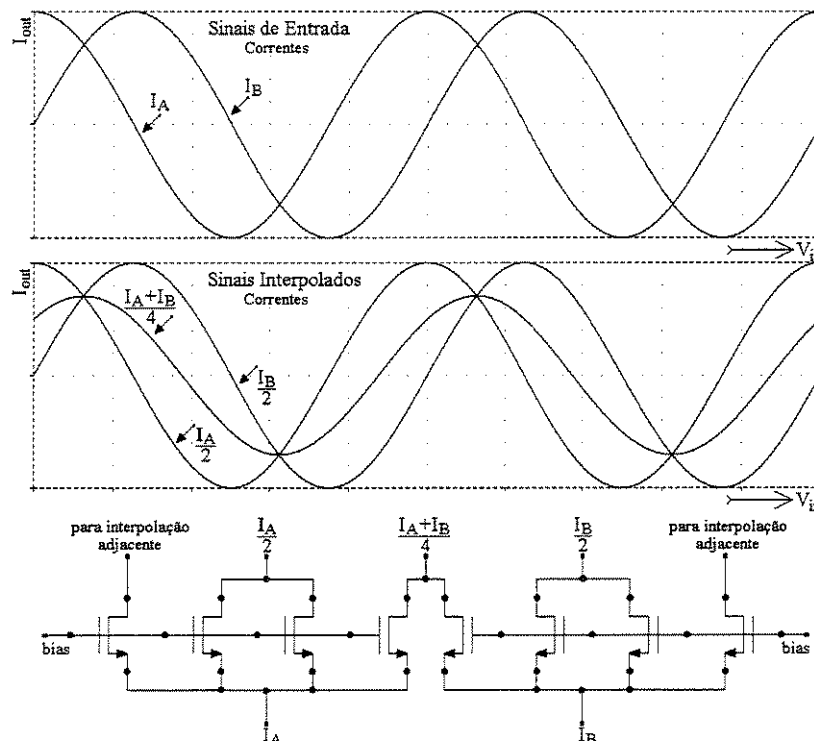


Figura 2.8 – Interpolação em Modo de Corrente

de correntes  $I_{\text{subl}}$  e  $I_{\text{subr}}$  subtraem parte da corrente de modo comum que vem dos sinais diferenciais  $I_{\text{fl}}$  e  $I_{\text{fr}}$ . Uma vez que o aumento da corrente afeta desfavoravelmente a precisão da interpolação, a maior parte da corrente de polarização é subtraída antes da divisão de corrente. Embora a subtração da corrente de polarização facilite a precisão necessária no divisor de corrente, as correntes subtraídas devem ter um determinado ajuste para se obter a precisão desejada. Considerando que os sinais não passam através de fontes de corrente, é possível utilizar transistores grandes com bom casamento entre si, sem que ocorra qualquer redução significativa da velocidade do circuito [11, 23 e 24]. As saídas dos “latches” são decodificadas para determinar os bits menos significativos.

Em contraste com a interpolação resistiva convencional, a monotonicidade não é garantida na interpolação por divisão de corrente. Portanto, o casamento entre os transistores deve ser cuidadosamente considerado. Os descasamentos das voltagens de “threshold” são geralmente pequenos e podem ser tratados como perturbação de pequeno sinal. Se a transcondutância do MOSFET para pequenos sinais for  $g_m$ , então a corrente que passa através de

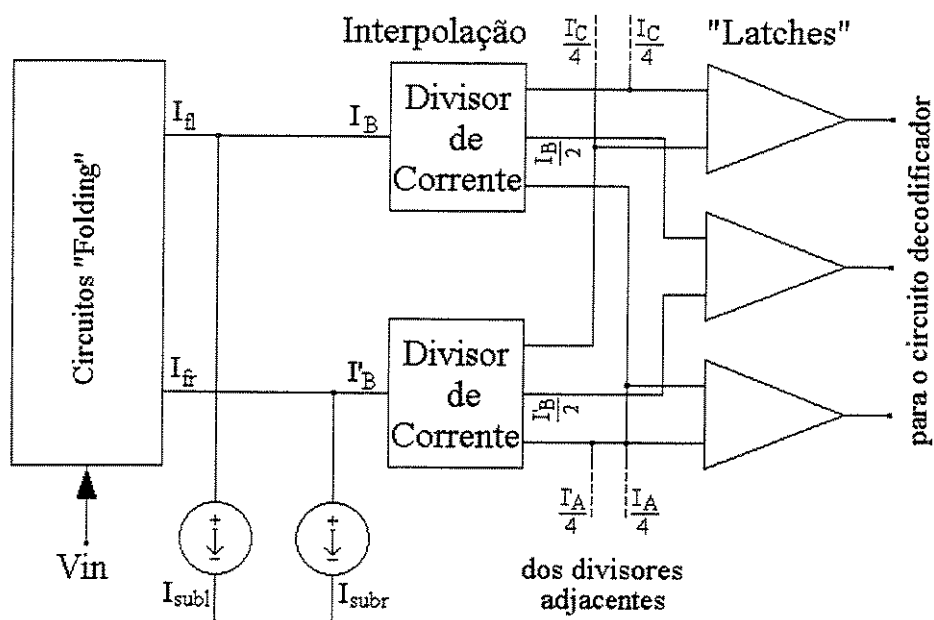


Figura 2.9 – Implementação da Interpolação em Modo de Corrente

cada transistor no divisor de corrente é igual a:

$$I_i = I_{\text{mean}} + g_m \cdot \Delta V_{Ti} \quad (2.1)$$

E a corrente  $I_n$  que passa através de “n” dos “N” transistores do divisor de corrente é:

$$I_n = n \cdot I_{\text{mean}} + g_m \cdot \sum_{i=1}^N \Delta V_{Ti} \quad (2.2)$$

E a corrente total,  $I_N$  é dada por:

$$I_N = N \cdot I_{\text{mean}} + g_m \cdot \sum_{i=1}^N \Delta V_{Ti} \quad (2.3)$$

O desvio padrão da corrente é dado pela equação abaixo:

$$\sigma(I_n) \propto \frac{\sqrt{I_D}}{L_{\text{Int}}} \quad (2.4)$$

Estas relações entre as correntes são dadas em [11]. Sendo que o desvio padrão da corrente é inversamente proporcional ao comprimento ( $L_{\text{Int}}$ ) do transistor, mas independe da largura, desde que a melhoria no casamento da voltagem de “threshold” ocorrida pelo aumento da largura seja cancelada pela diminuição da voltagem “porta-fonte” [11].

Conforme [16], esta técnica tem dois problemas: ela adiciona “nós” extras no caminho do sinal, reduzindo a largura de banda do conversor A/D e não trabalha bem com baixas voltagens de alimentação. Para evitar ambos os problemas, o circuito divisor de corrente deve ser integrado aos circuitos “folding”. Em [12], é dito que a maior desvantagem relativa da interpolação em modo de corrente em comparação com a interpolação resistiva é o aumento do consumo de potência.

De forma geral, nos conversores A/D tipo “folding”, com tecnologia CMOS, quando os sinais de saídas dos circuitos “folding” forem em voltagens, a interpolação deverá ser realizada com resistores e, quando os sinais de saída do circuitos “folding” forem em corrente, a interpolação deverá ser implementada com a técnica de divisão de corrente [16].

## 2.4 – Interpolação – Proposta Inicial:

Tratando-se de um conversor A/D de 6 bits tipo “folding” com interpolação, os três bits mais significativos são gerados diretamente pelos circuitos “folding” e os três bits menos significativos são determinados pela combinação apropriada de 8 sinais diferenciais quase senoidais (eles têm o mesmo nível, frequência e diferentes ângulos de fase). Os ângulos de fase destes sinais são mostrados na Fig. 2.10. Nesta figura, os sinais  $3SB/\overline{3SB}$  e  $S90/\overline{S90}$  (eles têm uma diferença de fase de  $90^\circ$ ) são determinados pelo circuito do “folding encoder” e os outros sinais são obtidos pela proposta inicial da nova técnica de interpolação. Estes 8 sinais diferenciais têm toda a informação necessária para determinar os 3 bits menos significativos (“fine” bits) do conversor A/D. Nesta proposta inicial da nova técnica de interpolação, os sinais  $S45/\overline{S45}$  e  $S135/\overline{S135}$  são obtidos em “latches” mestres simples (por este motivo chamada de interpolação simples) e os sinais  $S157.5/\overline{S157.5}$ ,  $S112.5/\overline{S112.5}$ ,  $S67.5/\overline{S67.5}$  e  $S22.5/\overline{S22.5}$  são obtidos em “latches” mestres duplos (por este motivo chamada de interpolação dupla).

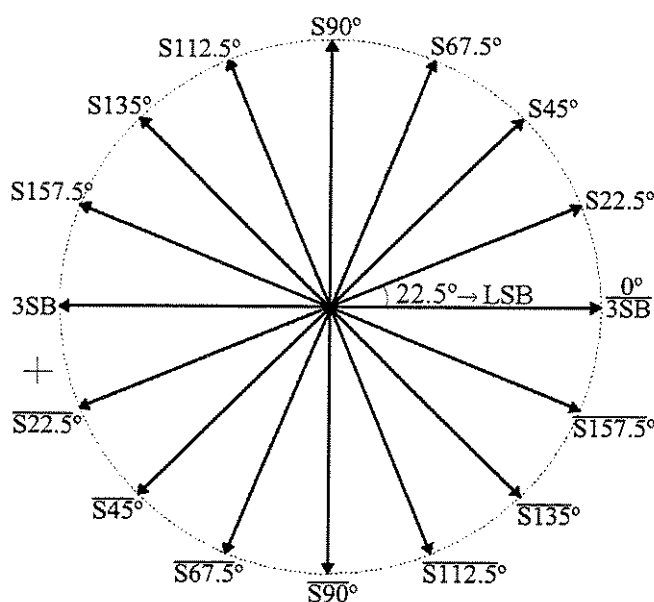


Figura 2.10 – Sinais Diferenciais de um Conversor A/D de 6 bits

### 2.4.1 – Interpolação Simples:

Na análise da técnica de interpolação simples, é admitido, para facilitar a análise, que o circuito pré-amplificador (no “latch” mestre simples) seja linear e com ganho unitário e tenha os sinais  $S90$  e  $3SB$  na entrada (veja a Fig. 2.11). Na saída, o sinal resultante é igual a  $[(S90)-(3SB)]$  e com ângulo de fase fixo de  $45^\circ$  em relação ao sinal  $\overline{3SB}$ . Admitindo-se os sinais interpolados como senoidais, tem-se na saída:

$$S90 = \text{sen}(\theta + 90^\circ) \text{ e } 3SB = \text{sen}(\theta + 180^\circ) \quad (2.5 \text{ e } 2.6)$$

$$\text{Em Out} \Rightarrow \text{sen}(\theta + 90^\circ) - \text{sen}(\theta + 180^\circ) \Rightarrow G \cdot \sqrt{2} \cdot \text{sen}(\theta + 45^\circ) \Rightarrow S45 \quad (2.7)$$

$$\text{Em } \overline{\text{Out}} \Rightarrow \text{sen}(\theta + 180^\circ) - \text{sen}(\theta + 90^\circ) \Rightarrow G \cdot \sqrt{2} \cdot \text{sen}(\theta + 225^\circ) = \overline{S45} \quad (2.8)$$

Na Fig. 2.11, é possível ver que o sinal resultante corresponde ao sinal  $S45$  (resultante de uma interpolação entre os sinais  $3SB$  e  $S90$ ). Sendo o circuito diferencial, este também tem o sinal  $\overline{S45}$ .

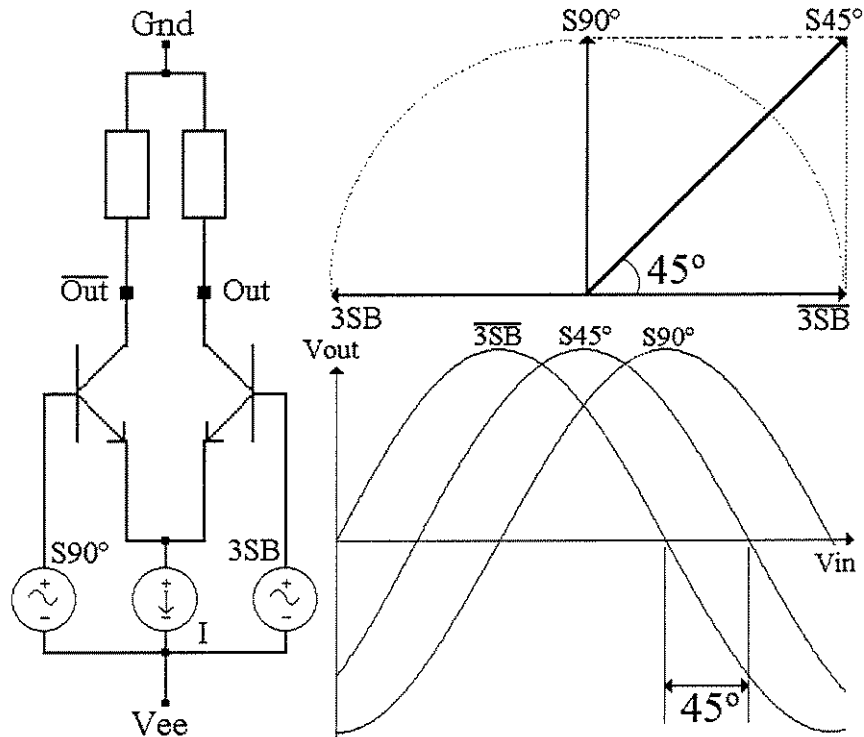


Figura 2.11 – Interpolação Simples

Realmente, o nível de saída dos sinais interpolado é uma função do ganho do circuito e tem uma pequena distorção, mas o cruzamento pelo zero está correto. Uma característica muito importante da interpolação simples é a alta insensibilidade à variação de temperatura porque o cruzamento de zero continua correto. Os sinais obtidos pela técnica de interpolação simples são:  $S_{45}/\overline{S_{45}}$  e  $S_{135}/\overline{S_{135}}$ . Os Sinais  $S_{135}/\overline{S_{135}}$  são obtidos da mesma forma pela operação  $[(S_{90})-(\overline{3SB})]$ .

#### 2.4.2 – Interpolação Dupla:

Como a interpolação simples permite obter apenas o ângulo de fase fixo de  $45^\circ$ , na obtenção dos demais sinais é necessário aplicar a chamada interpolação dupla (veja a Fig. 2.10). Na análise da interpolação dupla, procede-se da mesma maneira que na interpolação simples. Os sinais obtidos pela técnica da interpolação dupla são:  $S_{157.5}/\overline{S_{157.5}}$ ,  $S_{112.5}/\overline{S_{112.5}}$ ,  $S_{67.5}/\overline{S_{67.5}}$  e  $S_{22.5}/\overline{S_{22.5}}$ . Na Fig. 2.12, aparecem dois circuitos pré-amplificadores. Eles têm o mesmo consumo de potência (igual ao pré-amplificador simples, isto é,  $I = I_1 + I_2$ ) e são partes de um “latch” mestre duplo. A análise da técnica de interpolação dupla (válida para ambos os circuitos) é baseada na mesma figura 2.12. Da mesma forma que na análise da interpolação simples, é admitido, para facilitar a análise, que o circuito pré-amplificador duplo seja linear e com ganho unitário e tenha os sinais  $S_{90}$ ,  $3SB$  e  $\overline{3SB}$  na entrada. Na saída, o sinal resultante é igual a  $[(S_{90})-(3SB)]+[(\overline{3SB})-3SB]$  e com ângulo de fase  $\beta$  em relação ao sinal  $3SB$  (veja a Fig. 2.12). É possível demonstrar que  $\beta = \text{Arctag}\{1/[1+2.(I_2/I_1)]\}$ . A razão  $I_2/I_1$  é o parâmetro que controla  $\beta$ . No circuito 1 (veja a Fig. 2.12), as correntes dos circuitos pré-amplificadores são determinadas por fontes de correntes simples ( $I_1$  e  $I_2$ ). No circuito 2, existe somente uma fonte de corrente ( $I_1$ ) e as áreas de emissor dos transistores  $Q_1$  e  $Q_2$  podem ser feitas diferentes, o que irá causar uma razão diferente entre o valor das correntes para os dois pré-amplificadores. Ajustando as

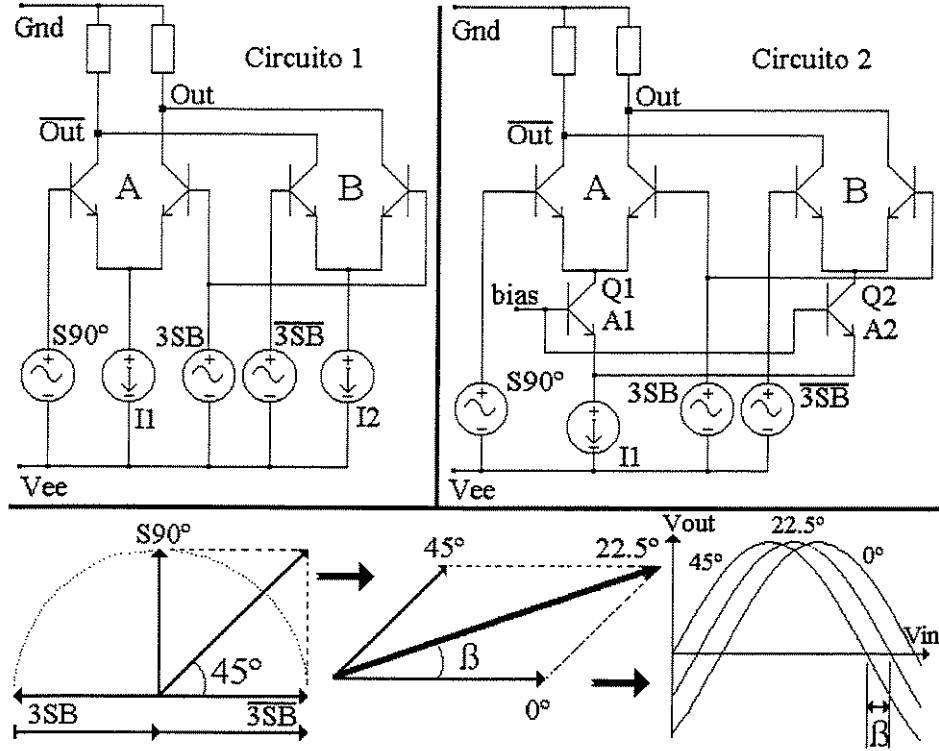


Figura 2.12 – Interpolação Dupla

correntes (ou as áreas dos emissores dos transistores) nos seguintes valores:  $I_2 = I_1 / \sqrt{2}$  (circuito 1) e  $A(Q_2) = A(Q_1) / \sqrt{2}$  (circuito 2), o sinal resultante na saída terá ângulo de fase de  $22,5^\circ$  em relação ao sinal  $\overline{3SB}$ . Entretanto, a exata razão entre as áreas do emissor dos transistores poderá necessitar de um pequeno ajuste durante a simulação do circuito. Admitindo-se os sinais interpolados como senoidais e a relação  $A(Q_2) = A(Q_1) / \sqrt{2}$  (desta forma, os sinais de saída dos dois pré-amplificadores têm o mesmo nível), tem-se na saída:

$$\text{Pré-amplificador "A"} \Rightarrow S90 = \sin(\theta + 90^\circ) \text{ e } 3SB = \sin(\theta + 180^\circ) \quad (2.9 \text{ e } 2.10)$$

$$\text{Em Out "A"} \Rightarrow \sin(\theta + 90^\circ) - \sin(\theta + 180^\circ) \Rightarrow G_a \cdot \sqrt{2} \cdot \sin(\theta + 45^\circ) \quad (2.11)$$

$$\text{Em } \overline{\text{Out}} \text{ "A"} \Rightarrow \sin(\theta + 180^\circ) - \sin(\theta + 90^\circ) \Rightarrow G_a \cdot \sqrt{2} \cdot \sin(\theta + 225^\circ) \quad (2.12)$$

$$\text{Pré-amplificador "B"} \Rightarrow \overline{3SB} = \sin(\theta + 0^\circ) \text{ e } 3SB = \sin(\theta + 180^\circ) \quad (2.13 \text{ e } 2.14)$$

$$\text{Em Out "B"} \Rightarrow \sin(\theta + 0^\circ) - \sin(\theta + 180^\circ) \Rightarrow G_b \cdot \sqrt{2} \cdot \sin(\theta + 0^\circ) (*) \quad (2.15)$$

$$\text{Em } \overline{\text{Out}} \text{ "B"} \Rightarrow \sin(\theta + 180^\circ) - \sin(\theta + 0^\circ) \Rightarrow G_b \cdot \sqrt{2} \cdot \sin(\theta + 180^\circ) (*) \quad (2.16)$$

(\*) Obs: com os níveis corrigidos.

$$\text{Em Out} \Rightarrow G_a \cdot \sqrt{2} \cdot \text{sen}(\theta + 45^\circ) + G_b \cdot \sqrt{2} \cdot \text{sen}(\theta + 0^\circ) \Rightarrow$$

$$G \cdot (\sqrt{2} \cdot \sqrt{2 + \sqrt{2}}) \cdot \text{sen}(\theta + 22,5^\circ) \Rightarrow S22.5 \quad (2.17)$$

$$\text{Em } \overline{\text{Out}} \Rightarrow G_a \cdot \sqrt{2} \cdot \text{sen}(\theta + 225^\circ) + G_b \cdot \sqrt{2} \cdot \text{sen}(\theta + 180^\circ) \Rightarrow$$

$$G \cdot (\sqrt{2} \cdot \sqrt{2 + \sqrt{2}}) \cdot \text{sen}(\theta + 202,5^\circ) \Rightarrow \overline{S22.5} \quad (2.18)$$

Na Fig. 2.12, é possível ver que o sinal resultante de saída corresponde ao sinal S22.5 (resultante da interpolação entre os sinais S90, 3SB e  $\overline{3SB}$ ). Sendo o circuito diferencial, este também tem o sinal  $\overline{S22.5}$ .

De fato, o nível de saída dos sinais interpolados é função do ganho do circuito e ele tem uma pequena distorção; mas o cruzamento de zero está correto. A variação do ângulo de fase dos sinais obtidos através da interpolação dupla é menor do que 0,1 LSB para uma variação de temperatura de 0°C a 70°C (para um conversor A/D de 6 bit). Outras interpolações para obter os sinais: S157.5/ $\overline{S157.5}$ , S112.5/ $\overline{S112.5}$  e S67.5/ $\overline{S67.5}$  são:

$$(S90 - \overline{3SB}) + (3SB - \overline{3SB}) \Rightarrow S157.5 \text{ e } \overline{S157.5}$$

$$(3SB - \overline{S90}) + (S90 - \overline{S90}) \Rightarrow S112.5 \text{ e } \overline{S112.5}$$

$$(\overline{3SB} - \overline{S90}) + (S90 - \overline{S90}) \Rightarrow S67.5 \text{ e } \overline{S67.5}$$

A característica da interpolação dupla é a variação do ângulo de fase resultante com a mudança na razão entre as correntes (ou áreas do emissor dos transistores). Desta forma, é possível obter diversos valores de ângulos de fase para os sinais interpolados.

#### 2.4.3 – Limitações:

Esta proposta inicial da nova técnica de interpolação evita a rede de interpolação resistiva e adiciona somente 3 transistores no “latch” mestre duplo e tem o mesmo consumo de potência porque a corrente é a mesma do “latch” mestre simples. Entretanto, devido à configuração

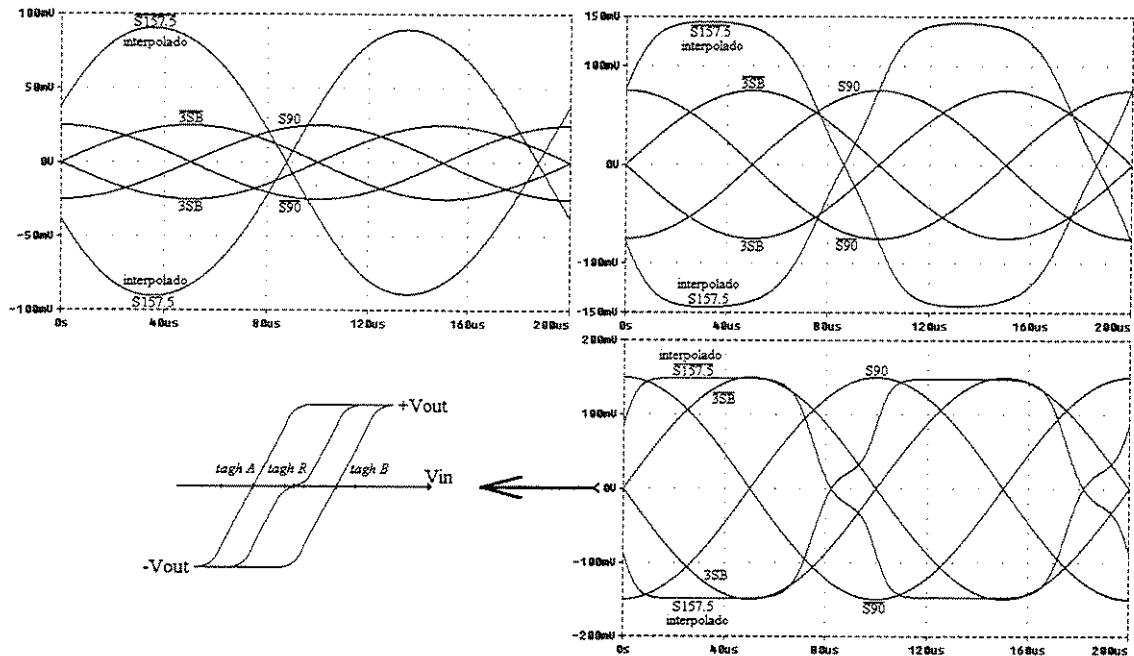


Figura 2.13 – *Distorção*

especial do circuito, alguns sinais nos “latches” mestre de interpolação não são diferenciais. Na análise anterior, foi suposto que o circuito pré-amplificador é linear. Realmente, o circuito recebe influência do nível do sinal de entrada porque, na prática, somente para diferenças de voltagens menores do que aproximadamente 50 mV ( $\sim 2V_T = 2.k.T/q$ ), o circuito pré-amplificador se comporta como em uma região aproximadamente linear. Quando o nível do sinal de entrada é próximo deste valor, não há distorção e o sinal de saída é quase perfeito. O incremento do nível do sinal de entrada resulta em uma pequena distorção no sinal interpolado. Isto não é problema porque o zero diferencial continua bem definido. Entretanto, se o nível do sinal de entrada (no “latch” duplo de interpolação) aumenta muito, ocorre uma grande distorção no sinal interpolado, porque o primeiro pré-amplificador deixa a região linear antes do segundo pré-amplificador ir para a região linear. A consequência é uma grande distorção no sinal de saída e o zero diferencial fica indefinido (veja a Fig. 2.13). Esta influência é maior quando os sinais de entrada tem uma diferença de fase maior do que 90°.

Uma dificuldade adicional é a perda da perfeita simetria bilateral do sinal, no ponto de

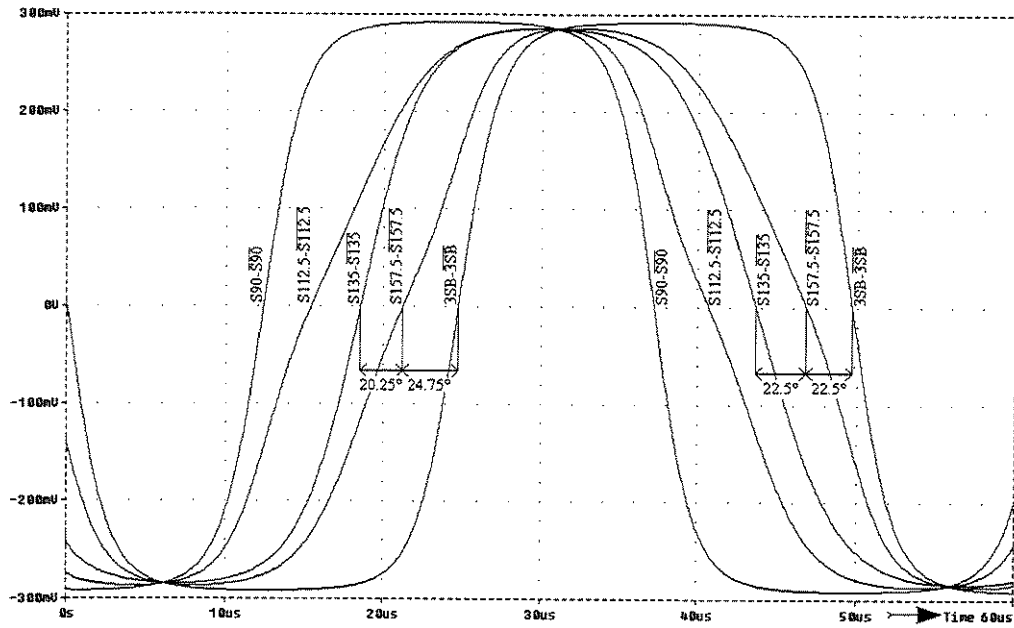


Figura 2.14 – Assimetria do Sinal

cruzamento do zero diferencial, com o aumento da distorção. Na Fig. 2.14 é possível verificar a existência da assimetria no cruzamento do zero diferencial. Os ângulos do sinal interpolado ficam diferentes nos pontos de cruzamento de zero diferencial, na subida e na descida do sinal. A diferença pode chegar a  $\sim 0,1$  LSB (ou  $\sim 2,25^\circ$  para um conversor A/D de 6 bits).

Embora os sinais interpolados tenham esta restrição, é possível obter uma larga soma de ângulos interpolados. De fato, para um nível de sinal de entrada além de 50 mV, eles devem ter uma diferença de fase menor (ou igual) a  $90^\circ$ . Por outro lado, o nível dos sinais de entrada não pode também ser muito baixo para não serem perturbados por ruídos e voltagem “offset” de

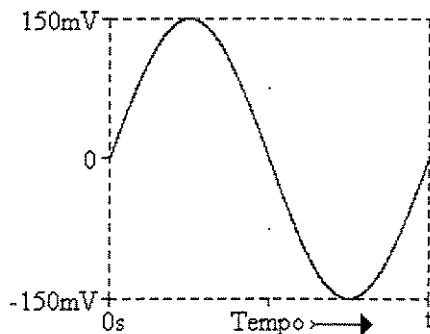


Figura 2.15 – Nível do Sinal

entrada (devido ao descasamento de  $V_{be}$  dos transistores). Portanto, o casamento dos transistores deve ser cuidadosamente considerado. De forma geral, devem-se utilizar sinais com níveis próximos de 300 mV (veja a Fig. 2.15).

Tabela 2.2 – *Ângulo  $\times$  LSB*

| Conversor A/D | A Diferença de $2,25^\circ$ Equivale a: |
|---------------|-----------------------------------------|
| 6 bits        | 0,1 LSB                                 |
| 7 bits        | 0,2 LSB                                 |
| 8 bits        | 0,4 LSB                                 |

#### 2.4.4 – Conclusão:

A proposta inicial da nova técnica de interpolação descrita provê outra alternativa com respeito a interpolação resistiva. Os resultados mostraram que a proposta de interpolação torna possível um conversor A/D com resolução de 6 bits. Para um conversor A/D de 8 bits as condições de ajuste do nível do sinal se tornam bastante críticas, principalmente no aspecto da simetria do sinal. Uma diferença de  $\sim 2,25^\circ$  na simetria do sinal, para um conversor A/D de 8 bits, corresponde a  $\sim 0,4$  LSB, o que prejudica significativamente a DNL (“differential nonlinearity”) e a INL (“integral nonlinearity”) (veja a Tabela 2.2).

#### 2.5 – Interpolação – Proposta Final:

Na seqüência desta pesquisa, foi estudada outra forma alternativa e mais aprimorada para se obter estes mesmos sinais interpolados que permite fazer um conversor A/D com resolução de até 8 bit.

Tratando-se de um conversor A/D de 8 bits tipo “folding” com interpolação, os três bits mais significativos são gerados diretamente pelos circuitos “folding” e os cinco bits menos significativos são determinados pela combinação apropriada de 32 sinais diferenciais quase senoidais (eles têm o mesmo nível, frequência e diferentes ângulos de fase). Os ângulos de fase destes sinais são mostrados na Fig. 2.16. Nesta figura, os sinais  $3SB/\overline{3SB}$ ,  $S135/\overline{S135}$ ,  $S90/\overline{S90}$  e  $S45/\overline{S45}$  (eles têm uma diferença de fase de  $45^\circ$ ) são obtidos no circuito do “folding encoder” e os demais sinais são obtidos por interpolação nos “latches” mestres. Estes 32 sinais diferenciais

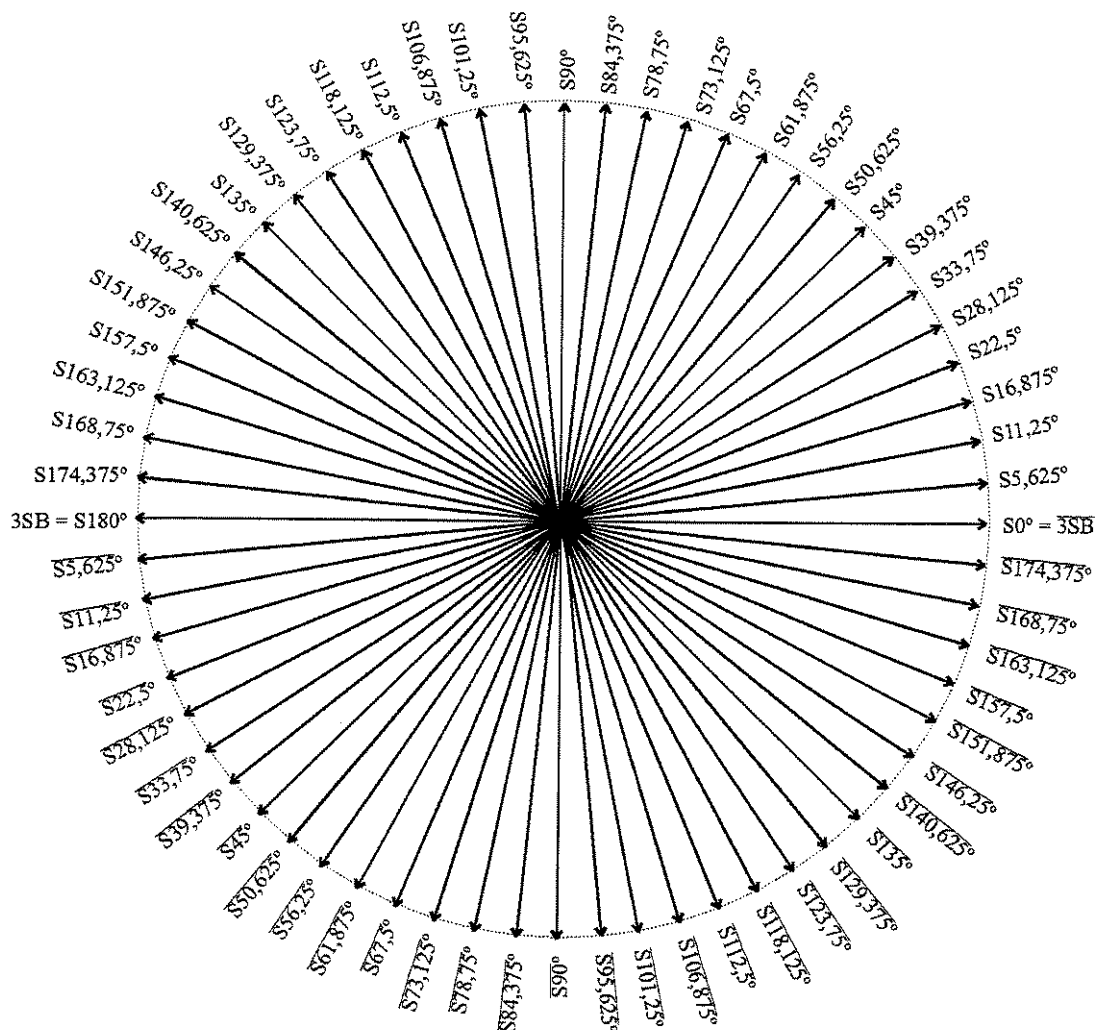


Figura 2.16 – Sinais Diferenciais de um Conversor A/D de 8 bits

têm toda a informação necessária para encontrar os 5 bits menos significativos (“fine” bits) do conversor A/D. Nesta proposta final da nova técnica de interpolação, os sinais  $S_{45}/\overline{S_{45}}$  e  $S_{135}/\overline{S_{135}}$  são obtidos por interpolação dos sinais de saída dos amplificadores “folding” no próprio “folding encoder” (e foi chamada de interpolação simples) e os demais sinais interpolados são obtidos por interpolação nos “latches” mestres duplos (por este motivo chamada de interpolação dupla).

### 2.5.1 – Interpolação Simples:

No capítulo anterior foi visto que é possível combinar no “folding encoder” os sinais de

saída do amplificador “folding” para obter os sinais  $MSB$ ,  $\overline{MSB}$ ,  $2SB$ ,  $\overline{2SB}$ ,  $3SB$ ,  $\overline{3SB}$ ,  $S90$  e  $\overline{S90}$ ; também é possível, como será demonstrado mais adiante, obter os sinais  $S45$ ,  $\overline{S45}$ ,  $S135$  e  $\overline{S135}$ . Desta maneira, a interpolação simples é feita no próprio circuito do “folding encoder”. Os sinais  $S45$ ,  $\overline{S45}$ ,  $S135$  e  $\overline{S135}$  são obtidos por combinação (e interpolação) dos sinais oriundos dos amplificadores “folding” 1 e 2. Veja, a seguir, as combinações dos sinais das saídas dos amplificadores “folding” no “folding encoder” e compare as Fig. 2.17, 2.19 e 2.20. A Fig. 2.18 mostra o circuito no “folding encoder” para a obtenção dos sinais  $S45/\overline{S45}$  e  $S135/\overline{S135}$ .

$$S45 \Rightarrow (2A + 4A + 6A + 8A + 10A + 1B + 3B + 5B + 7B + 9B + 11B)$$

$$\overline{S45} \Rightarrow (1A + 3A + 5A + 7A + 9A + 2B + 4B + 6B + 8B + 10B)$$

$$S135 \Rightarrow (1A + 3A + 5A + 7A + 9A + 1B + 3B + 5B + 7B + 9B + 11B)$$

$$\overline{S135} \Rightarrow (2A + 4A + 6A + 8A + 10A + 2B + 4B + 6B + 8B + 10B)$$

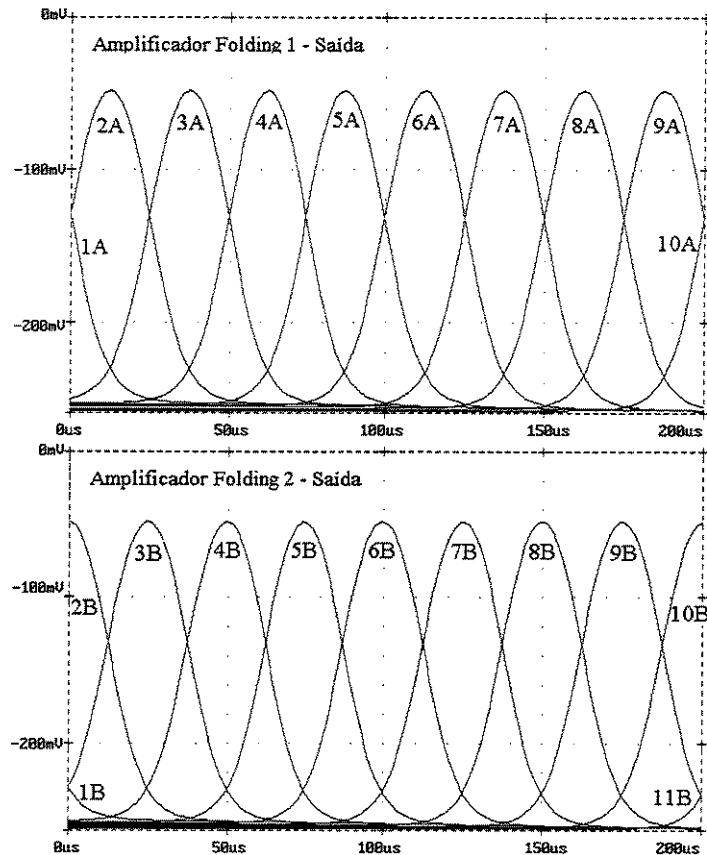


Figura 2.17 – Sinais de Saída dos Amplificadores “Folding”

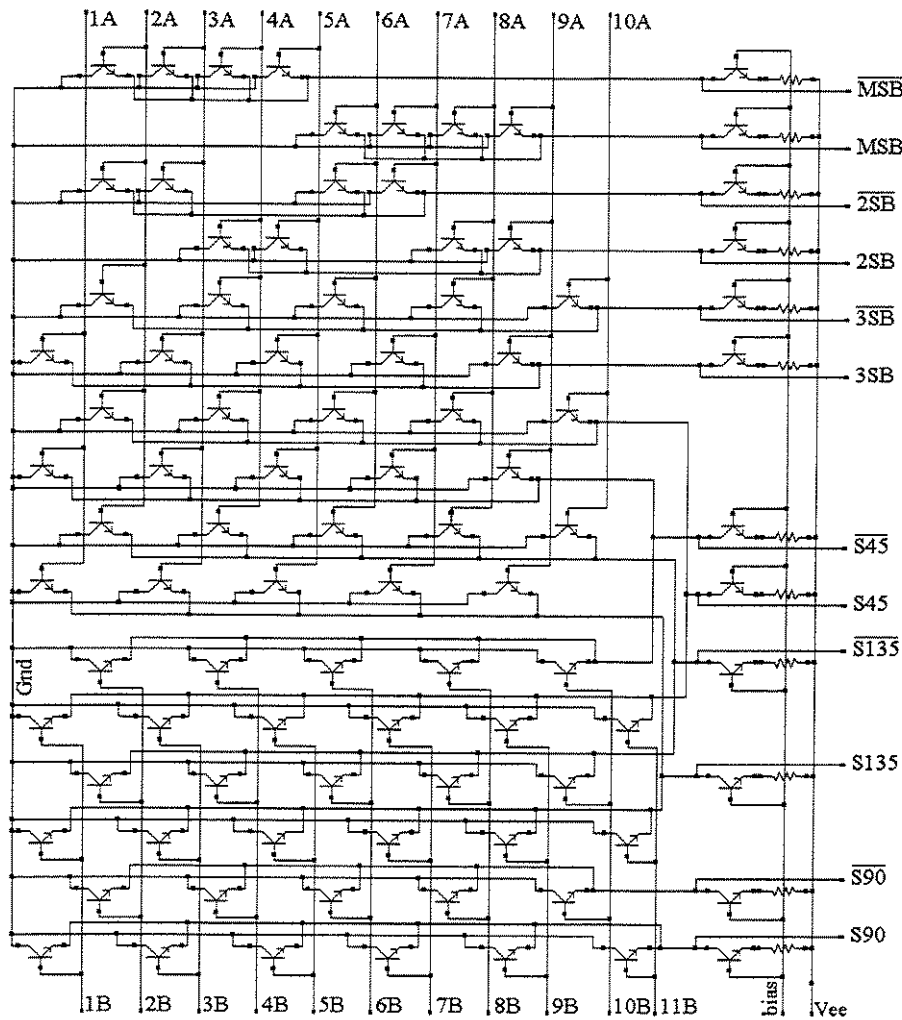


Figura 2.18 – Circuito no “Folding Encoder” para os sinais  $S45/\overline{S45}$  e  $S135/\overline{S135}$

As Fig. 2.19 e 2.20 mostram a formação dos sinais  $S45/\overline{S45}$  e  $S135/\overline{S135}$ , sendo possível perceber que estes sinais somente podem ser utilizados no modo diferencial. Na Fig. 2.21 é possível ver os sinais interpolados ( $S45/\overline{S45}$  e  $S135/\overline{S135}$ ) em relação aos sinais  $3SB/\overline{3SB}$  e  $S90/\overline{S90}$ . Duas características muito importantes da interpolação simples são: a inexistência de distorção no sinal interpolado (em relação aos sinais  $3SB/\overline{3SB}$  e  $S90/\overline{S90}$ ) e a alta insensibilidade à variação de temperatura porque o cruzamento de zero continua correto (veja a Fig. 2.22). A vantagem adicional é a possibilidade de usar estes sinais no próximo nível de interpolação (que nós chamamos de interpolação dupla nos “latches” mestre duplos) e a desvantagem é o pequeno

incremento no consumo (de aproximadamente 2 mW, pelo aumento do circuito do “folding encoder”) o que reduz um pouco a eficiência no consumo de potência do conversor A/D.

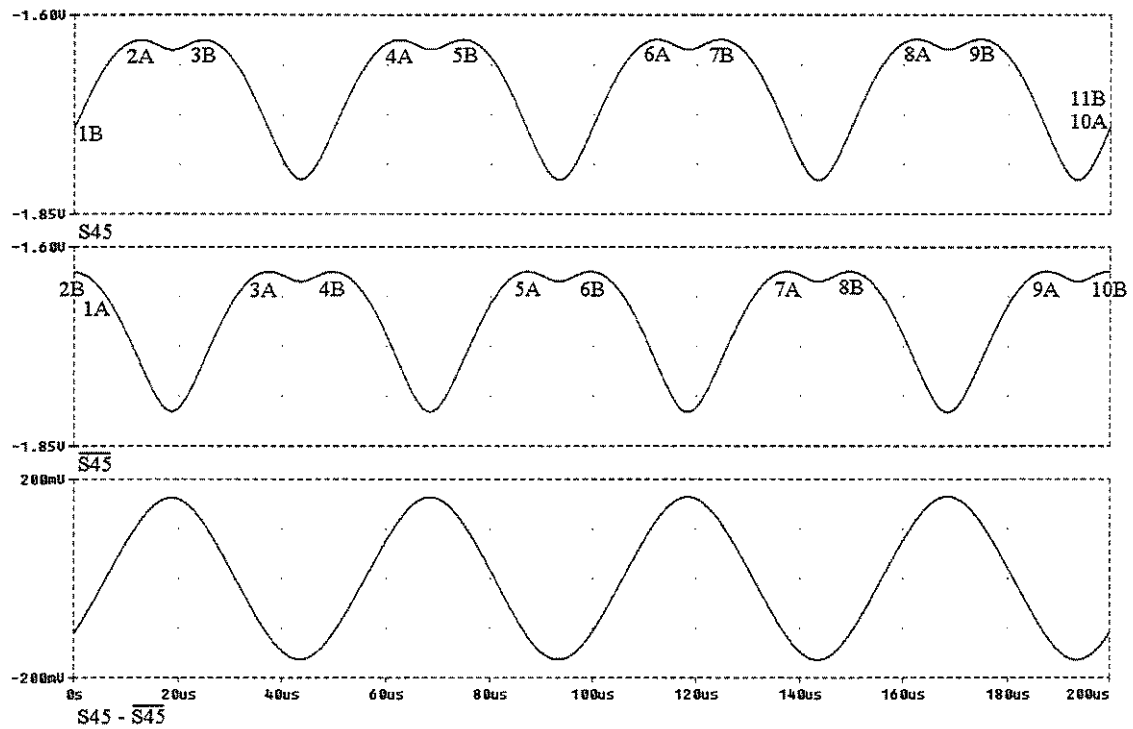


Figura 2.19 – Sinal  $S45/\overline{S45}$

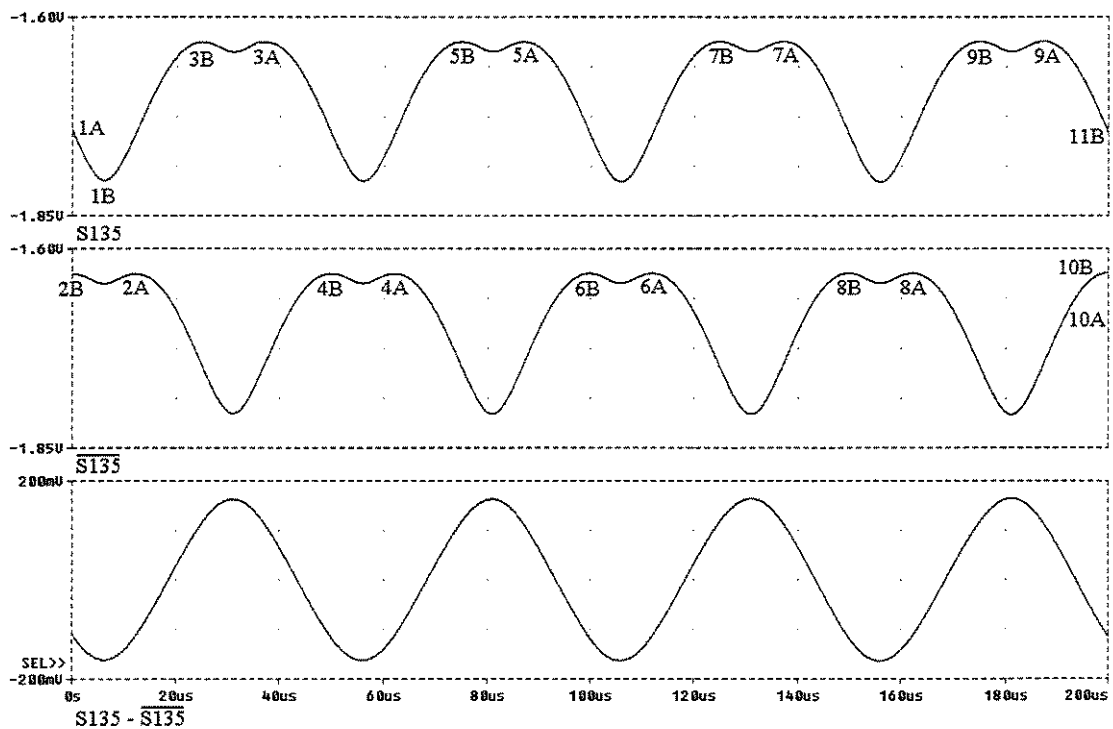


Figura 2.20 – Sinal  $S135/\overline{S135}$

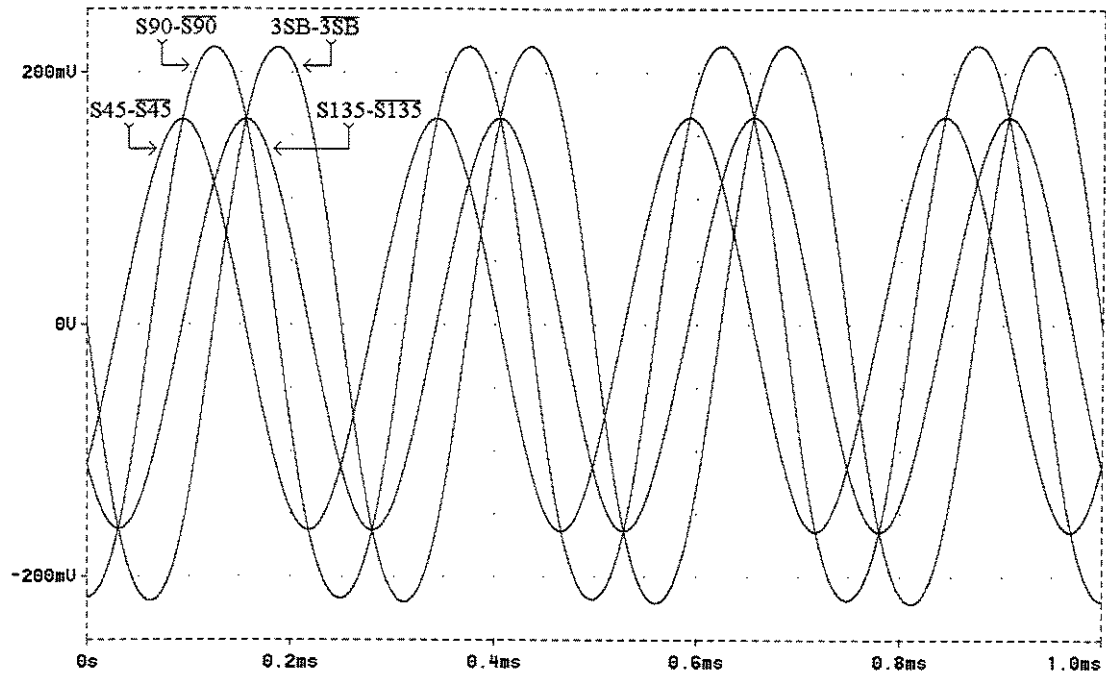


Figura 2.21 – Sinais ( $S45 - \overline{S45}$ ), ( $S135 - \overline{S135}$ ), ( $3SB - \overline{3SB}$ ) e ( $S90 - \overline{S90}$ )

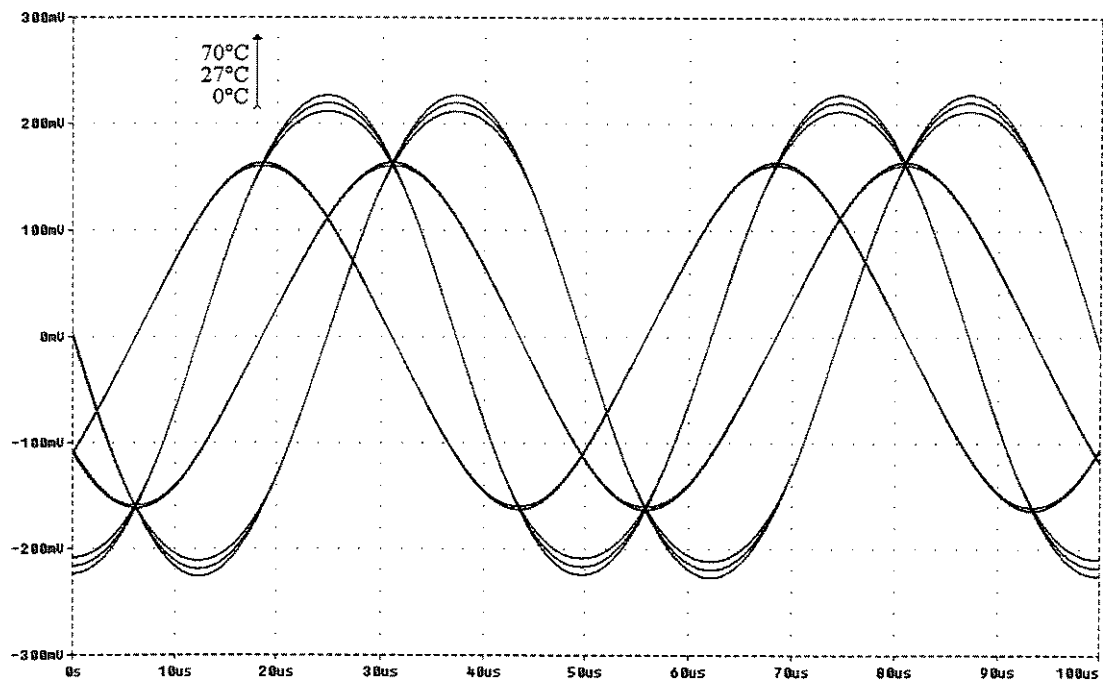


Figura 2.22 – Variação dos Sinais com a Temperatura

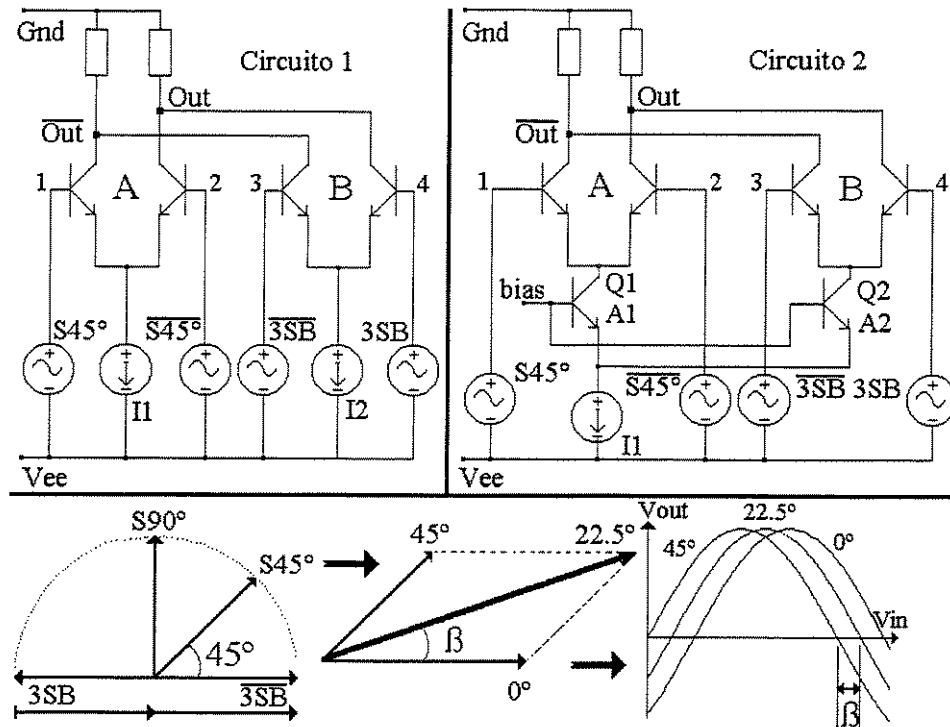


Figura 2.23 – Interpolação Dupla

### 2.5.2 – Interpolação Dupla:

Na Fig. 2.16, exceto os sinais  $3SB/\overline{3SB}$ ,  $S135/\overline{S135}$ ,  $S90/\overline{S90}$  e  $S45/\overline{S45}$ , todos os demais são obtidos através da interpolação dupla. Os circuitos utilizados pela interpolação dupla na proposta final são os mesmos utilizados pela interpolação dupla na proposta inicial. O que muda radicalmente é a aplicação dos sinais na entrada do “latch” de interpolação dupla porque todos os sinais aplicados passam a ser diferenciais. Na Fig. 2.23, são mostrados dois circuitos pré-amplificadores. Eles têm o mesmo consumo de potência (igual ao pré-amplificador simples, isto é,  $I = I_1 + I_2$ ) e são partes de um “latch” mestre duplo. A análise da técnica de interpolação dupla (válida para ambos os circuitos) é baseada na mesma Fig. 2.23. Para facilitar a análise, é admitido que o circuito pré-amplificador duplo é linear e com ganho unitário.

Existem quatro formas básicas de aplicar o sinal na entrada dos circuitos mostrados na Fig. 2.23 e todos os sinais interpolados (nos “latches” mestres duplos) são resultantes destas

Tabela 2.3 – *Sinais Aplicados e os Sinais Gerados por Interpolação*

| Entradas: 1 / 2 / 3 / 4                           | Saída dos Sinais Resultantes da Interpolação                                                                                                                                                                                        |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S45 / $\overline{S45}$ / $\overline{3SB}$ / 3SB   | S5.625/ $\overline{S5.625}$ , S11.25/ $\overline{S11.25}$ , S16.875/ $\overline{S16.875}$ , S22.5/ $\overline{S22.5}$ , S28.125/ $\overline{S28.125}$ , S33.75/ $\overline{S33.75}$ , S39.375/ $\overline{S39.375}$                 |
| S45 / $\overline{S45}$ / S90 / $\overline{S90}$   | S50.625/ $\overline{S50.625}$ , S56.25/ $\overline{S56.25}$ , S61.875/ $\overline{S61.875}$ , S67.5/ $\overline{S67.5}$ , S73.125/ $\overline{S73.125}$ , S78.75/ $\overline{S78.75}$ , S84.375/ $\overline{S84.375}$               |
| S135 / $\overline{S135}$ / S90 / $\overline{S90}$ | S95.625/ $\overline{S95.625}$ , S101.25/ $\overline{S101.25}$ , S106.875/ $\overline{S106.875}$ , S112.5/ $\overline{S112.5}$ , S118.125/ $\overline{S118.125}$ , S123.75/ $\overline{S123.75}$ , S129.375/ $\overline{S129.375}$   |
| S135 / $\overline{S135}$ / 3SB / $\overline{3SB}$ | S140.625/ $\overline{S140.625}$ , S146.25/ $\overline{S146.25}$ , S151.875/ $\overline{S151.875}$ , S157.5/ $\overline{S157.5}$ , S163.125/ $\overline{S163.125}$ , S168.75/ $\overline{S168.75}$ e S174.375/ $\overline{S174.375}$ |

ligações com as respectivas razões entre as correntes (ou áreas do emissor) dos transistores. A Tabela 2.3 mostra as quatro ligações básicas e os sinais interpolados possíveis de serem gerados.

Para exemplificar, será analisado o comportamento dos circuitos da Fig. 2.23 (válida para ambos os circuitos), supondo-se que os sinais S45,  $\overline{S45}$ ,  $\overline{3SB}$  e 3SB sejam aplicados nas entradas 1, 2, 3 e 4, respectivamente. Como as amplitudes dos sinais diferenciais S45/ $\overline{S45}$  e  $\overline{3SB}$ /3SB são diferentes (veja a Fig. 2.21 e a Fig. 1.9 no capítulo anterior), para o sinal interpolado resultante ter uma fase de 22,5° (S22.5/ $\overline{S22.5}$ ), a razão entre as correntes deve ser de  $1/\sqrt{2}$  ou seja:

$$I_2 = \frac{I_1}{\sqrt{2}} \rightarrow I_1 = I_2 \cdot \sqrt{2} \quad (\text{para o circuito 1}) \quad (2.19)$$

ou

$$A(Q_2) = \frac{A(Q_1)}{\sqrt{2}} \rightarrow A(Q_1) = A(Q_2) \cdot \sqrt{2} \quad (\text{para o circuito 2}) \quad (2.20)$$

Desta forma, a influência (o peso) de cada um dos sinais a serem interpolados fica equivalente e o sinal resultante da interpolação terá 22,5°. Isto é, na saída, o sinal resultante será igual a [(S45)-( $\overline{S45}$ )]+[( $\overline{3SB}$ )-3SB] e com ângulo de fase  $\beta = 22,5^\circ$  em relação ao sinal  $\overline{3SB}$  (veja a Fig. 2.23). A amplitude do sinal resultante não apresenta nenhum problema, porque como

todos os sinais (interpolados ou não) passam por “latches” mestres, as saídas são todas normalizadas para um nível digital. É possível demonstrar que o ângulo de fase “ $\beta$ ” do sinal resultante é:

$$\beta = \text{Arctag}[(G_A \cdot \sin 45^\circ + G_B \cdot \sqrt{2} \cdot \sin 0^\circ) / (G_A \cdot \cos 45^\circ + G_B \cdot \sqrt{2} \cdot \cos 0^\circ)] \quad (2.21)$$

$$= \text{Arctag}[G_A / (G_A + 2 \cdot G_B)] = \text{Arctag}[g_{mA} / (g_{mA} + 2 \cdot g_{mB})] = \text{Arctag}[I_1 / (I_1 + 2 \cdot I_2)]$$

$$\text{Portanto: } \beta = \text{Arctag} \{1/[1+2 \cdot (I_2/I_1)]\} \Rightarrow \text{para o circuito 1 da Fig. 2.23.} \quad (2.22)$$

$$\text{Sendo: } \frac{I_1}{I_2} = \frac{A(Q_1)}{A(Q_2)}, \text{ tem-se:}$$

$$\beta = \text{Arctag} \{1/[1+2 \cdot (A(Q_2)/A(Q_1))]\} \Rightarrow \text{para o circuito 2 da Fig. 2.23.} \quad (2.23)$$

A razão  $I_2/I_1$  [ou  $A(Q_2)/A(Q_1)$ ] é o parâmetro que controla  $\beta$ . Como todos os sinais a serem interpolados possuem uma diferença de fase igual a  $45^\circ$  e, também, a diferença de amplitude dos sinais é semelhante nas quatro formas básicas de interpolação, a equação (2.22 ou 2.23) é válida para todas as quatro formas básicas de aplicação dos sinais interpoladores nos “latches” mestres duplos.

No circuito 1 (veja a Fig. 2.23), as correntes dos circuitos pré-amplificadores são determinadas por fontes de correntes simples ( $I_1$  e  $I_2$ ). No circuito 2, existe somente uma fonte de corrente ( $I_1$ ) e as áreas do emissor dos transistores  $Q_1$  e  $Q_2$  podem ser feitas diferentes, o que irá causar uma razão diferente entre o valor das correntes para os dois pré-amplificadores. A Tabela 2.4 mostra os sinais interpoladores aplicados aos “latches” mestres duplos, os sinais interpolados resultantes e as razões utilizadas de  $I_2/I_1$  [ou  $A(Q_2)/A(Q_1)$ ].

A característica da interpolação dupla é a variação do ângulo de fase resultante com a troca da razão entre as correntes (ou áreas do emissor dos transistores). Desta forma, é possível obter diversos valores de ângulos de fase para os sinais interpolados.

Tabela 2.4 – Fase dos Sinais Aplicados e as Relações de  $I_2/I_1$  ou  $A(Q_2)/A(Q_1)$ 

| S45/S45<br>e<br>3SB/3SB | S45/S45<br>e<br>S90/S90 | S135/S135<br>e<br>S90/S90 | S135/S135<br>e<br>3SB/3SB | Razão<br>$I_2/I_1$ ou<br>$A(Q_2)/A(Q_1)$ |
|-------------------------|-------------------------|---------------------------|---------------------------|------------------------------------------|
| 39,375°                 | 50,625°                 | 129,375°                  | 140,625°                  | 0,11                                     |
| 33,75°                  | 56,25°                  | 123,75°                   | 146,25°                   | 0,25                                     |
| 28,125°                 | 61,875°                 | 118,125°                  | 151,875°                  | 0,44                                     |
| 22,5°                   | 67,5°                   | 112,5°                    | 157,5°                    | 0,71                                     |
| 16,875°                 | 73,125°                 | 106,875°                  | 163,125°                  | 1,15                                     |
| 11,25°                  | 78,75°                  | 101,25°                   | 168,75°                   | 2,01                                     |
| 5,625°                  | 84,375°                 | 95,625°                   | 174,375°                  | 4,58                                     |

De fato, o sinal interpolado possui uma distorção mas o cruzamento de zero está correto. A variação do ângulo de fase dos sinais obtidos através da interpolação dupla é menor do que 0,3 LSB para uma variação de temperatura de 0°C a 70°C (para um conversor A/D de 8 bit) (veja a Fig. 2.24 e 2.25).

### 2.5.3 – Limitações:

A proposta final da nova técnica de interpolação também evita a rede de interpolação resistiva e adiciona 46 transistores no circuito “folding encoder” e somente 3 transistores em cada um dos “latch” mestre duplo. Na interpolação simples ocorre um pequeno incremento no consumo de potência em função apenas do aumento do circuito de “folding encoder”. A parte relativa à interpolação dupla não provoca aumento no consumo de potência porque a corrente do “latch” mestre duplo é a mesma do “latch” mestre simples. Devido à configuração especial do circuito, todos os sinais utilizados nos “latches” mestre duplo de interpolação são diferenciais.

A proposta final de interpolação possui a mesma limitação existente na proposta inicial, em termos de nível dos sinais de entrada. Esta limitação será recapitulada a seguir. Na análise desta proposta de interpolação, foi suposto que o circuito pré-amplificador é linear. Realmente, o circuito recebe influência do nível do sinal de entrada porque, na prática, somente para diferenças de voltagens menores do que aproximadamente 50 mV ( $\sim 2V_T = 2.k.T/q$ ), o circuito pré-

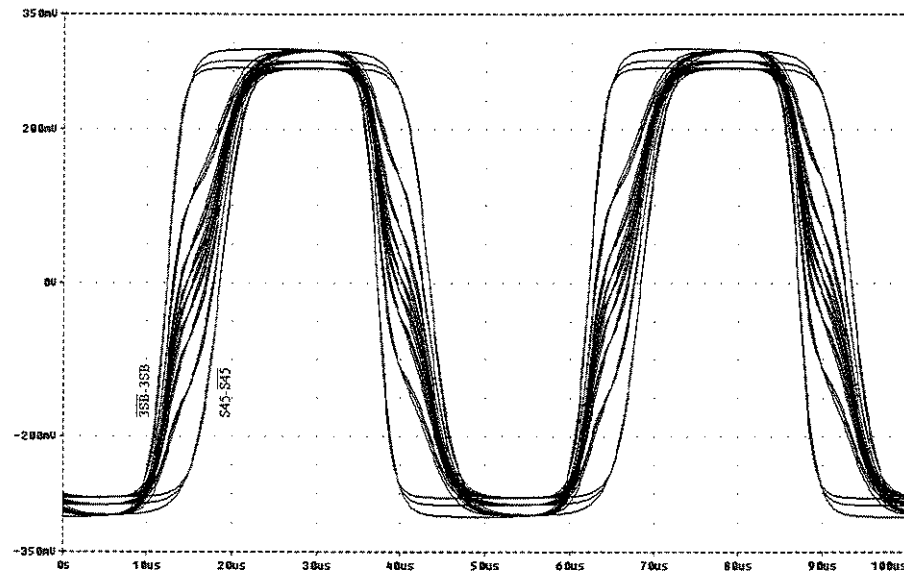


Figura 2.24 – *Variação, com a Temperatura, dos Sinais Interpolados*

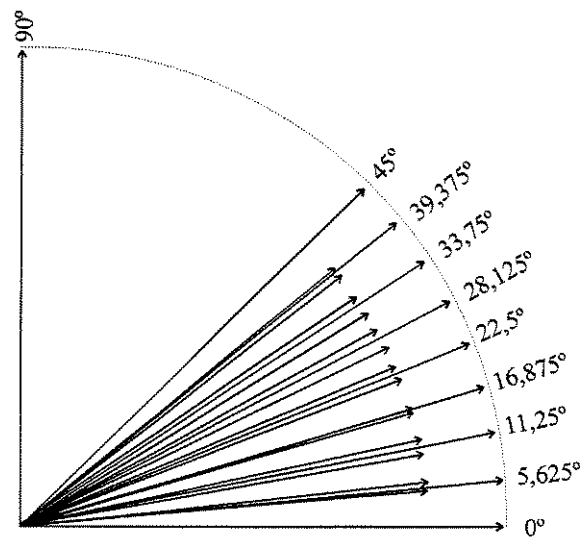


Figura 2.25 – *Variação, com a Temperatura, da Fase dos Sinais Interpolados*

amplificador se comporta como em uma região aproximadamente linear.

Quando o nível do sinal de entrada está próximo deste valor, não há distorção e o sinal de saída é quase perfeito. O incremento do nível do sinal de entrada resulta em uma pequena distorção no sinal interpolado (veja a Fig. 2.26). Isto não é problema porque o zero diferencial continua bem definido. Entretanto, se o nível do sinal de entrada (no “latch” duplo de interpolação) aumenta muito, ocorre uma grande distorção no sinal interpolado porque o primeiro

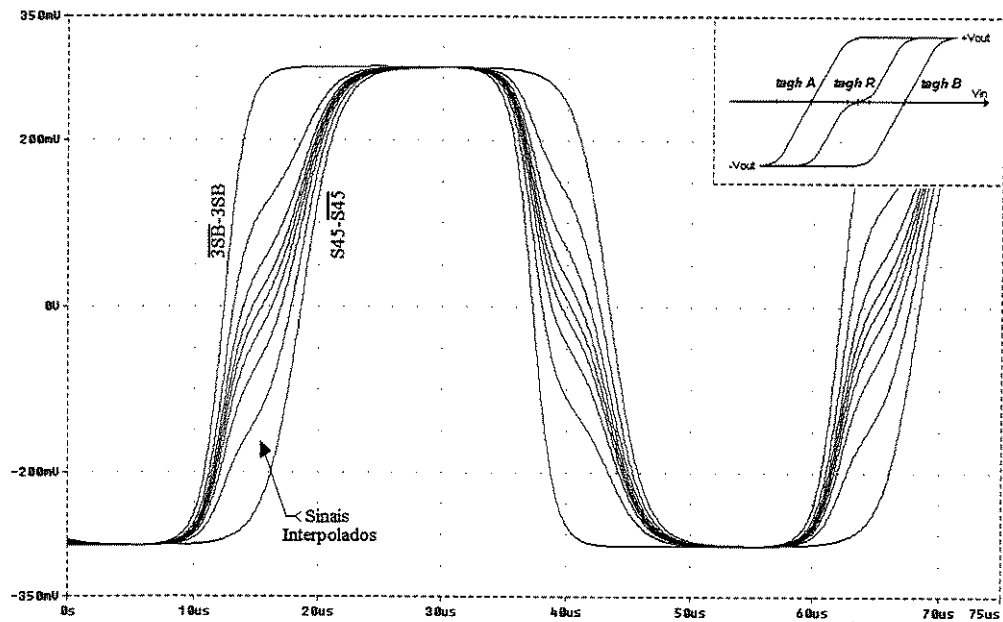


Figura 2.26 – *Distorção nos Sinais Interpolados*

pré-amplificador deixa a região linear antes do segundo pré-amplificador ir para a região linear. A consequência é uma grande distorção no sinal de saída e o zero diferencial fica indefinido (veja o quadro menor na Fig. 2.26). Na proposta final de interpolação, esta influência fica menor porque todos os sinais de entrada são diferenciais e têm uma diferença de fase igual a  $45^\circ$ .

Apesar do sinal resultante apresentar uma pequena distorção, ele possui uma perfeita simetria bilateral, no ponto de cruzamento do zero diferencial (veja um exemplo na Fig. 2.27). Os ângulos do sinal interpolado ficam iguais nos pontos de cruzamento de zero diferencial, na subida e na descida do sinal.

Para garantir o comportamento térmico adequado dos sinais utilizados para fazer a interpolação dupla ( $3SB/\overline{3SB}$ ,  $S135/\overline{S135}$ ,  $S90/\overline{S90}$  e  $S45/\overline{S45}$ ), é necessário a utilização de uma fonte de corrente tipo “PTAT” (“proportional to absolute temperature”) na polarização (“bias”) do circuito do amplificador “folding” e do circuito do “folding encoder”.

O nível dos sinais de entrada também não pode ser muito baixo para não ser perturbado por ruídos e tensão “offset” de entrada (devido ao descasamento de  $V_{be}$  dos transistores).

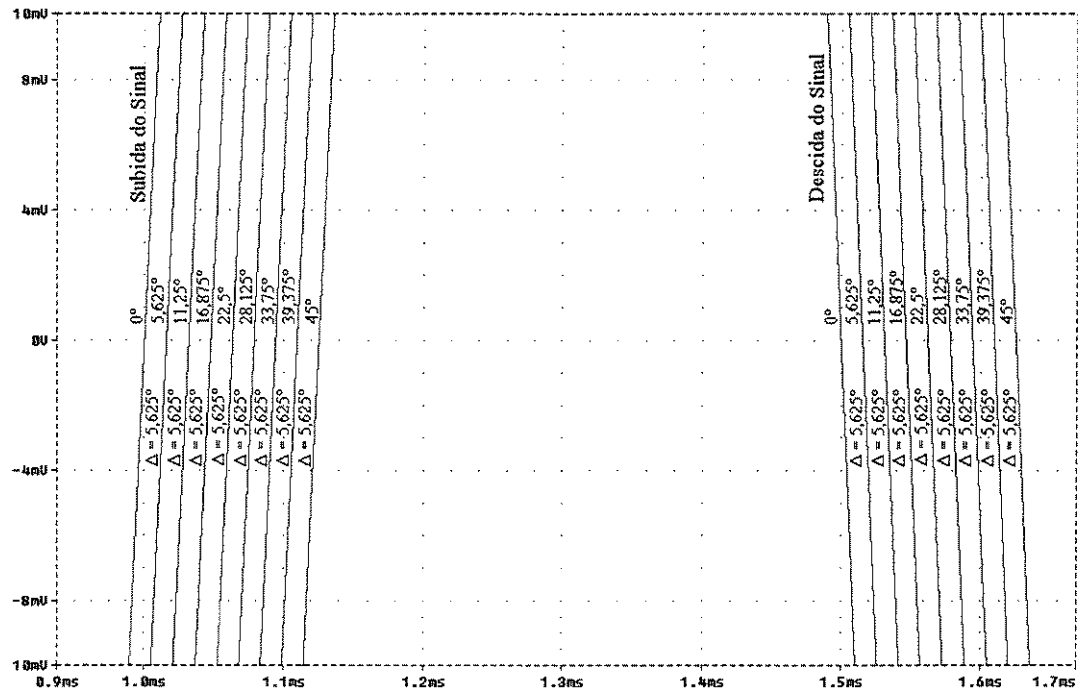


Figura 2.27 – Exemplo da Simetria Necessária nos Sinais Interpolados

Desta forma, o casamento dos transistores deve ser cuidadosamente considerado. De forma geral, devem-se utilizar sinais diferenciais com níveis próximos de 300 mV. Embora os sinais interpolados tenham esta restrição, é possível obter todos os sinais interpolados necessários para um conversor de 8 bits.

#### 2.5.4 – Conclusão:

A proposta final da nova técnica de interpolação, anteriormente descrita, é uma alternativa ao uso da interpolação resistiva. Os resultados mostraram que esta proposta de interpolação torna possível um conversor A/D com resolução de 8 bits, desde que cuidados especiais tenham sido aplicados na determinação das áreas dos transistores de interpolação (nos “latches” mestres duplos de interpolação). O problema de assimetria do sinal foi resolvido, o que permite a implementação de um conversor A/D de 8 bits com valores padrões aceitos para a DNL (“differential nonlinearity”) e para a INL (“integral nonlinearity”).

## Capítulo 3

### CONVERSOR ANALÓGICO/DIGITAL DE 8 BITS - PROJETO

#### 3.1 – Características Gerais:

Foi projetado um conversor A/D de 8 bits utilizando-se a técnica “folding” com interpolação. Esta resolução foi adotada porque é a mais utilizada nos conversores A/D rápidos (com alta taxa de conversão) e permite demonstrar perfeitamente a técnica “folding”, a proposta final da nova técnica de interpolação e, ainda, foram acrescentadas mais duas características inovadoras neste conversor A/D. A primeira é uma nova técnica para correção dos chamados erros de “bolha” (“bubble error”) e a segunda é uma nova topologia na configuração do circuito de conversão na técnica “folding”, o que permite diminuir o número total de componentes no circuito, com a correspondente diminuição no consumo de potência. O conversor A/D também possui o circuito de sincronismo de bits, que corrige os erros de temporização nos bits mais significativos, antes que eles sejam enviados para a saída. Neste conversor A/D, os 3 primeiros bits (MSB, 2SB e 3SB) são gerados diretamente pelo amplificador “folding” e “folding encoder” e os 5 últimos bits (4SB, 5SB, 6SB, 7SB e LSB) são obtidos através da técnica de interpolação. Desta forma, manteve-se a razão de folding em 8. A alimentação é de  $-5\text{ V}$ , a voltagem de referência ( $V_{\text{ref}}$ ) é de  $-1,125\text{ V}$ , o sinal de entrada ( $V_{\text{in}}$ ) é de  $1\text{ V}$  (variando de  $-1,0625\text{ V}$  a  $-0,0625\text{ V}$ ) e os sinais de saída são em código binário com nível ECL 10 000. O diagrama de blocos do conversor A/D pode ser visto na Fig. 3.1 e o circuito pode ser visto na Fig. 3.2.

#### 3.2 – Amplificadores “Folding”:

Verificando o diagrama de blocos da Fig. 3.1, percebe-se que o sinal de entrada ( $V_{\text{in}}$ ) é aplicado nos dois amplificadores “folding”. O primeiro amplificador “folding” possui 8 estágios

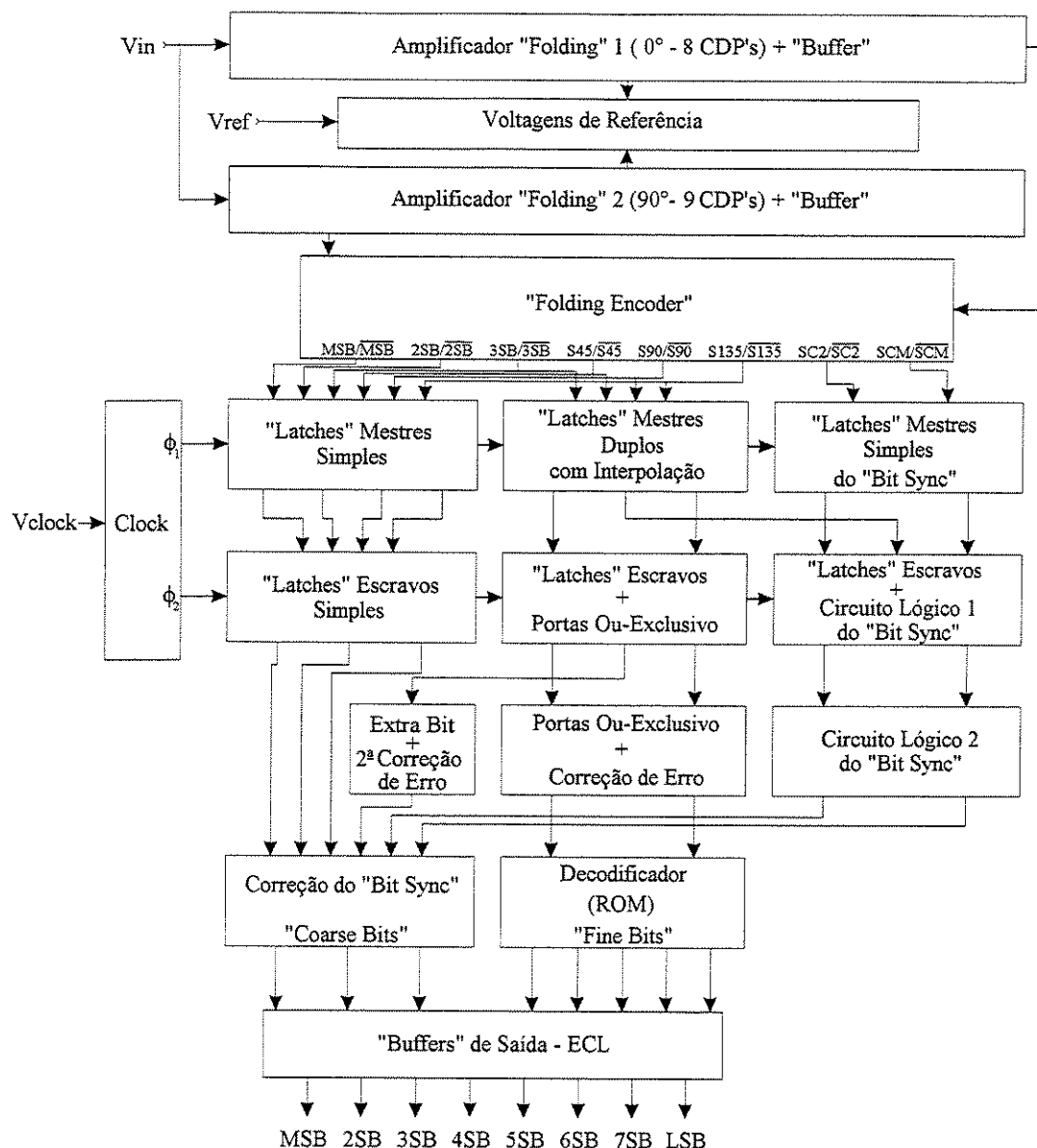


Figura 3.1 – Diagrama de Blocos do Conversor A/D de 8 Bits

pré-amplificadores combinados (também chamados de CDP [07]) e possui um “buffer” em cada uma das saídas. O segundo amplificador “folding”, para a correta elaboração da sua curva de saída, possui 9 estágios pré-amplificadores combinados (CDP) e também possui um “buffer” em cada uma das saídas. A Fig. 3.3 mostra o circuito básico utilizado para o CDP e a Fig. 3.4 mostra o circuito completo do amplificador “folding” que foi utilizado no conversor A/D.

As tensões de referência para os amplificadores “folding” são obtidas através de um divisor resistivo (veja a Fig. 3.5). Na prática, utilizou-se  $V_{ref} = -1,125\text{ V}$  e foram acrescentados

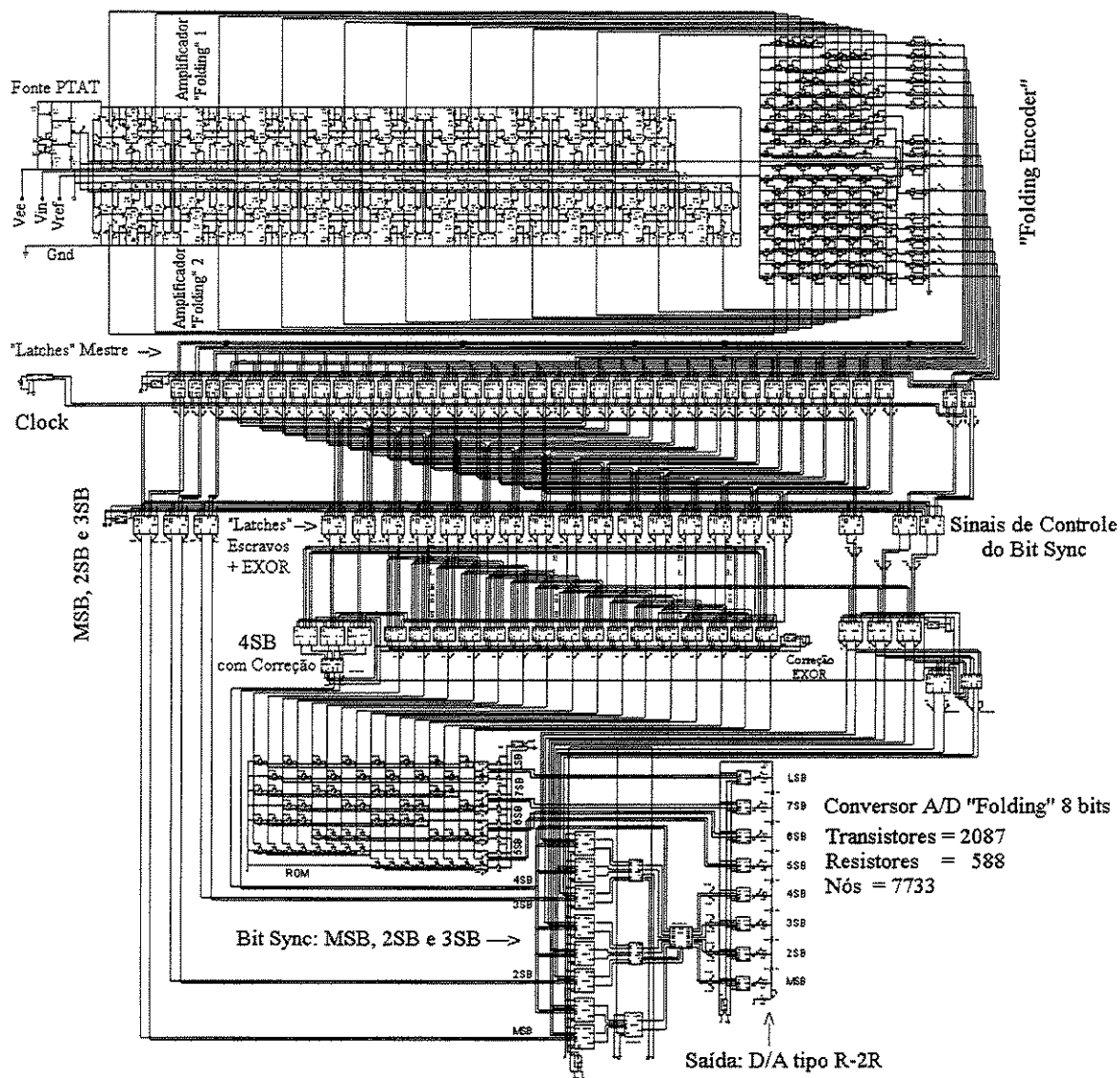


Figura 3.2 – Circuito do Conversor A/D de 8 Bits

mais dois resistores na rede resistiva utilizada como referência. Isto permite ajustar corretamente as formas das ondas geradas, porém, causou o "offset" de 62,5 mV do sinal analógico de entrada ( $V_{in}$ ). Em função da referência de voltagem, os amplificadores "folding" geram formas de ondas semelhantes, porém, defasadas de 90°. Existe um "buffer" em cada uma das entradas ( $V_{in}$  e  $V_{ref}$ ) e na saída ( $V_{out}$ ) de cada CDP.

Para um sinal analógico de entrada ( $V_{in}$ ) igual ao mostrado na Fig. 3.6 (tipo rampa), os sinais de saída dos dois amplificadores "folding" são mostrados na Fig. 3.7.

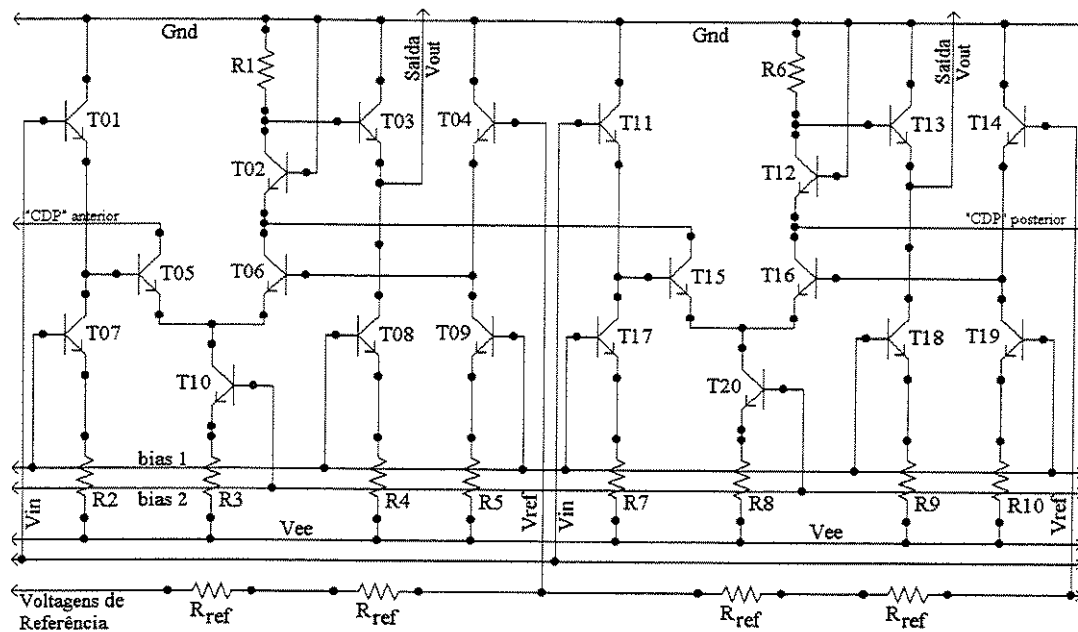


Figura 3.3 – CDP (“Coupled Differential Pair”)

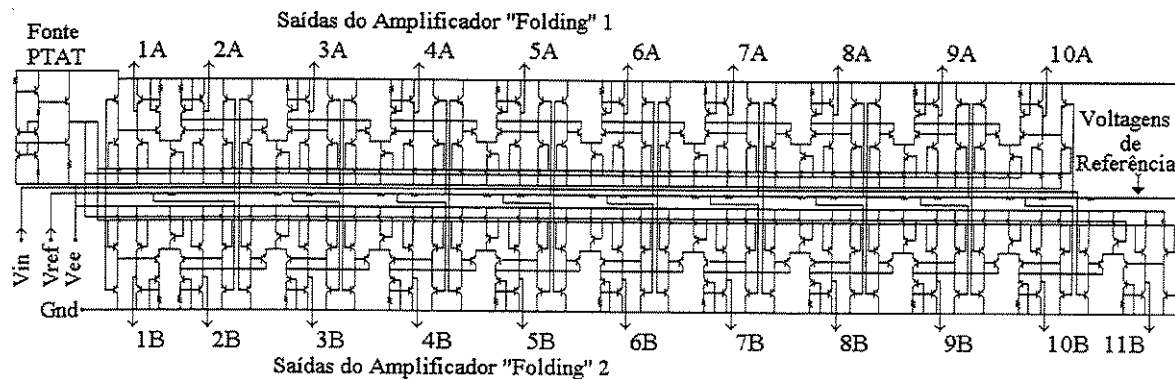


Figura 3.4 – Amplificadores “Folding”

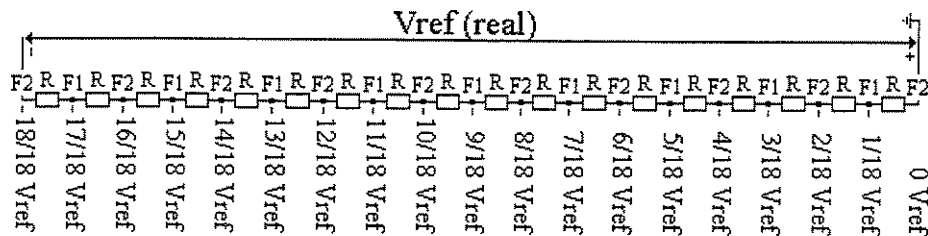


Figura 3.5 – Circuito de Referência de Voltagem

### 3.3 – “Folding Encoder”:

O circuito do “folding encoder” é uma estrutura tipo “ROM” que combina os sinais de saída dos amplificadores “folding” em 16 sinais complementares (veja a Fig. 3.8). Estes sinais são: MSB,  $\overline{\text{MSB}}$ , 2SB,  $\overline{2\text{SB}}$ , 3SB,  $\overline{3\text{SB}}$ , S45,  $\overline{\text{S45}}$ , S90,  $\overline{\text{S90}}$ , S135,  $\overline{\text{S135}}$ , SC2,  $\overline{\text{SC2}}$ , SCM e  $\overline{\text{SCM}}$ .

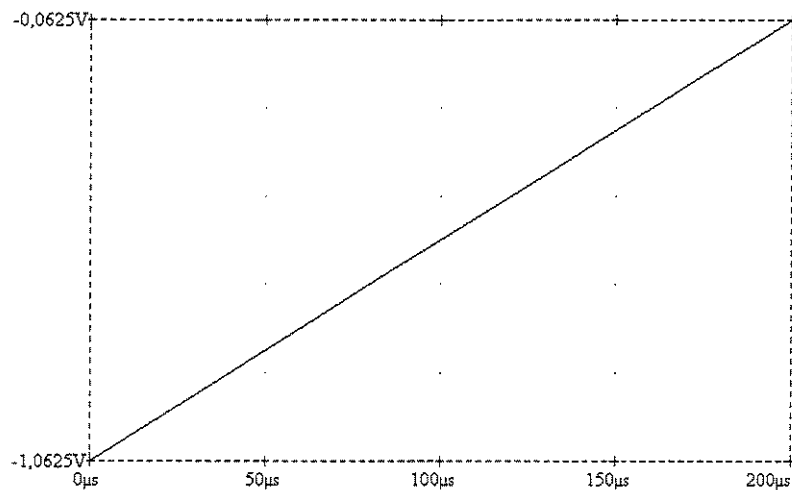


Figura 3.6 – Sinal de Entrada do Conversor A/D ( $V_{in}$ )

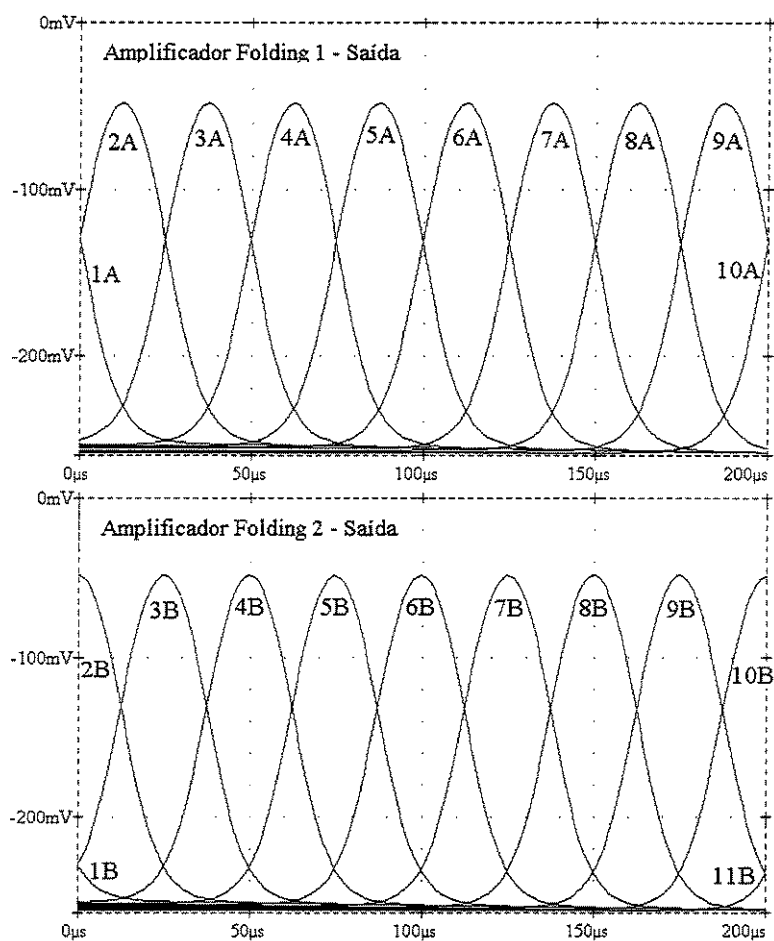


Figura 3.7 – Sinais de Saída dos Amplificadores “Folding”

Veja, a seguir, as combinações dos sinais das saídas do amplificador “folding” no “folding encoder”, para a obtenção destes sinais:

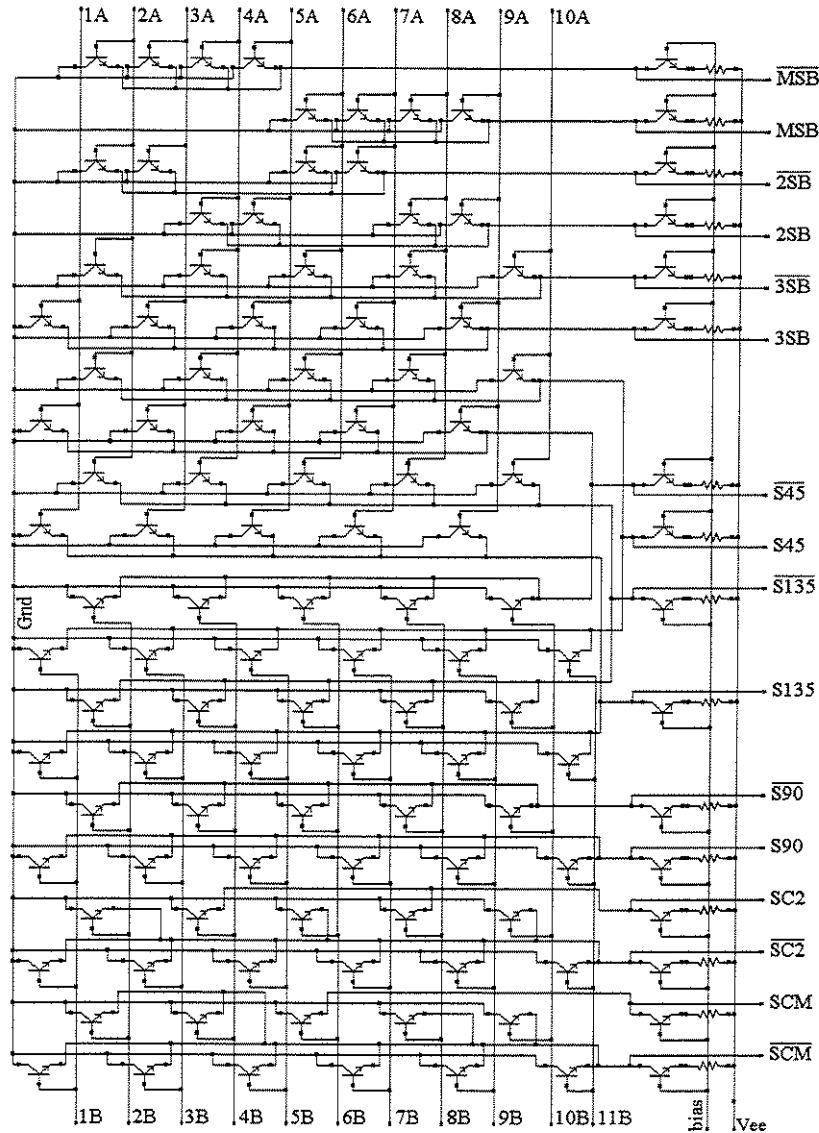


Figura 3.8 – Circuito do “Folding Encoder”

$$\text{MSB} \Rightarrow (6A + 7A + 8A + 9A) \text{ e } \overline{\text{MSB}} \Rightarrow (2A + 3A + 4A + 5A)$$

$$2\text{SB} \Rightarrow (4A + 5A + 8A + 9A) \text{ e } \overline{2\text{SB}} \Rightarrow (2A + 3A + 6A + 7A)$$

$$3\text{SB} \Rightarrow (1A + 3A + 5A + 7A + 9A) \text{ e } \overline{3\text{SB}} \Rightarrow (2A + 4A + 6A + 8A + 10A)$$

$$\text{S90} \Rightarrow (1B + 3B + 5B + 7B + 9B + 11B) \text{ e } \overline{\text{S90}} \Rightarrow (2B + 4B + 6B + 8B + 10B)$$

$$\text{S45} \Rightarrow (2A + 4A + 6A + 8A + 10A + 1B + 3B + 5B + 7B + 9B + 11B)$$

$$\overline{\text{S45}} \Rightarrow (1A + 3A + 5A + 7A + 9A + 2B + 4B + 6B + 8B + 10B)$$

$$\text{S135} \Rightarrow (1A + 3A + 5A + 7A + 9A + 1B + 3B + 5B + 7B + 9B + 11B)$$

$$\overline{\text{S135}} \Rightarrow (2A + 4A + 6A + 8A + 10A + 2B + 4B + 6B + 8B + 10B)$$

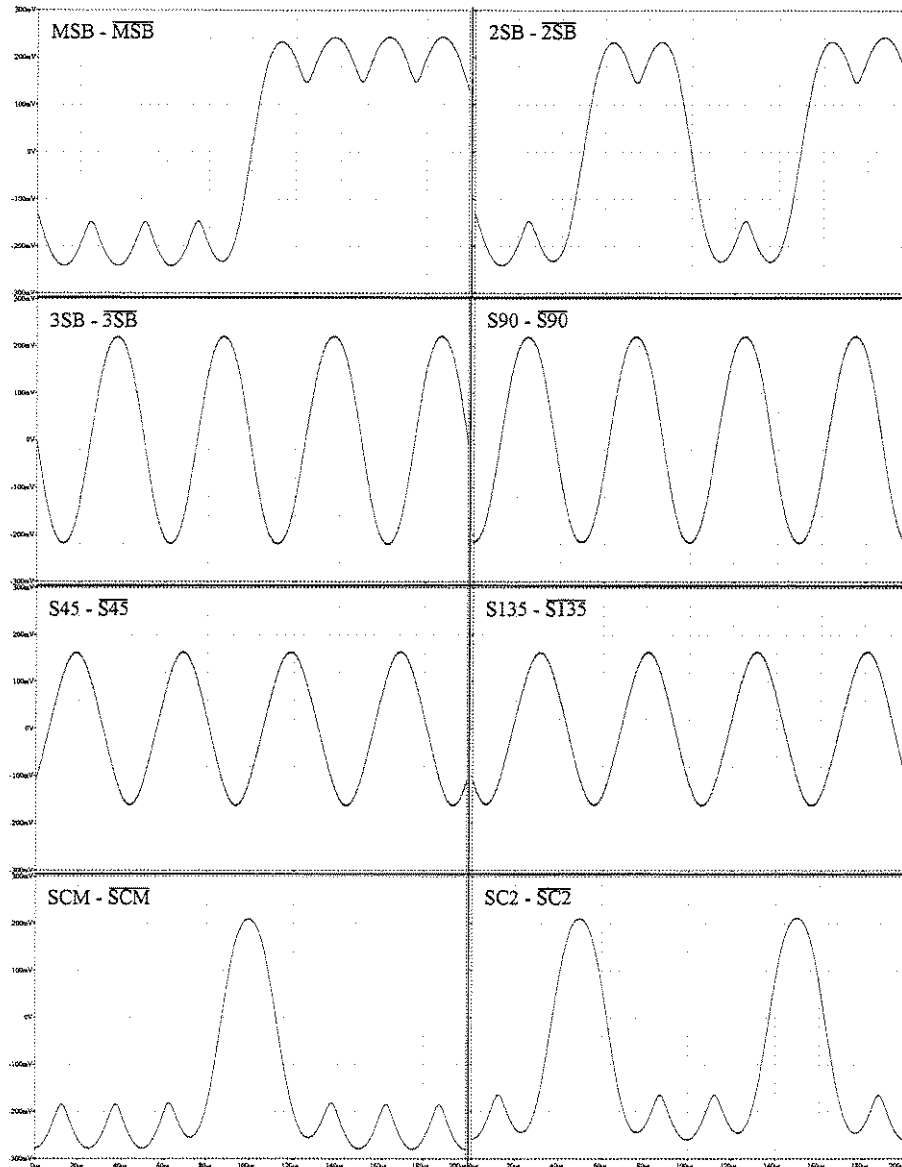


Figura 3.9 – Sinais de Saída do “Folding Encoder”

$SCM \Rightarrow (6B)$  e  $\overline{SCM} \Rightarrow (1B + 2B + 3B + 4B + 5B + 6B + 7B + 8B + 9B + 10B + 11B)$

$SC2 \Rightarrow (4B + 8B)$  e  $\overline{SC2} \Rightarrow (1B + 2B + 3B + 5B + 6B + 7B + 9B + 10B + 11B)$

Os sinais  $MSB$ ,  $\overline{MSB}$ ,  $2SB$ ,  $\overline{2SB}$ ,  $3SB$  e  $\overline{3SB}$  são obtidos por combinação dos sinais oriundos do amplificador “folding” 1 e os sinais  $S90$  e  $\overline{S90}$ , de forma semelhante, são obtidos por combinação dos sinais oriundos do amplificador “folding” 2.

Os sinais  $S45$ ,  $\overline{S45}$ ,  $S135$  e  $\overline{S135}$  são obtidos por combinação (e interpolação) dos sinais

oriundos dos amplificadores “folding” 1 e 2. Estes sinais são obtidos através da chamada interpolação simples (da proposta final), demonstrada na capítulo anterior.

Os sinais SCM,  $\overline{\text{SCM}}$ , SC2 e  $\overline{\text{SC2}}$  são obtidos por combinação dos sinais oriundos do amplificador “folding” 2 e são necessários para o controle do sincronismo de bits (“Bit Sync”).

A Fig. 3.9 mostra todos estes sinais na forma diferencial. A ondulação observada sobre os sinais (MSB-  $\overline{\text{MSB}}$ ) e (2SB- $\overline{2\text{SB}}$ ) é inerente a sua formação, ou seja, é o resultado da soma dos sinais dos amplificadores “folding” e não acarreta nenhum problema porque irá desaparecer quando o sinal passar pelo “latches” mestre do conversor A/D. Isto é, como todos os sinais passarão pelos “latches” mestre, ficando as saídas normalizadas em um nível digital.

### 3.4 – Fonte de Polarização:

Para garantir o comportamento térmico adequado dos sinais, no circuito do amplificador “folding” e do circuito do “folding encoder”, é necessário a utilização de uma fonte de corrente tipo “PTAT” na polarização (“bias”). A Fig. 3.10 mostra a fonte de polarização utilizada, que

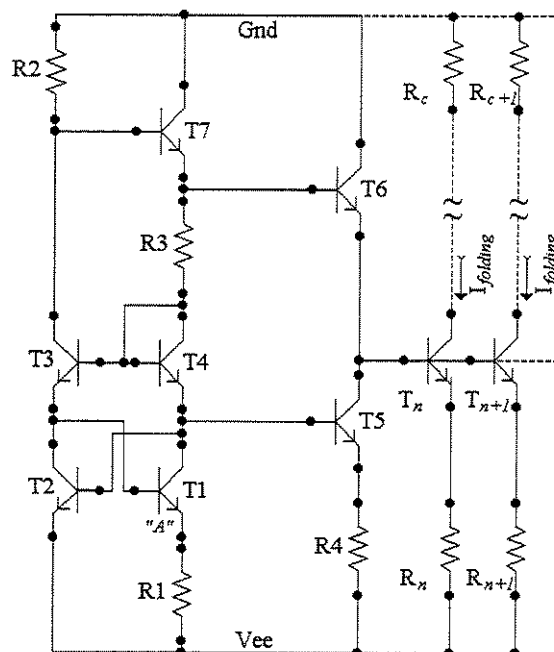


Figura 3.10 – Fonte de Polarização

utiliza somente transistores NPN; foi baseada em uma fonte apresentada em [08] e que se caracteriza pela simplicidade.

As Fig. 3.11 (a) e 3.11 (b) mostram um exemplo do comportamento do sinal (3SB-3SB) sem e com o uso da fonte de polarização tipo PTAT respectivamente. É perceptível que com a fonte de polarização tipo PTAT, o sinal sofre uma menor variação. A Fig. 3.12 mostra o comportamento da corrente ( $I_{\text{folding}}$ ) com a variação da temperatura (variação de 0°C a 70°C).

A corrente ( $I_{\text{folding}}$ ) é igual a:

$$I_{\text{folding}} = \frac{V_{be2} + V_{be4} - V_{be6} - V_{be_n}}{R_n} + \frac{R_3 \cdot k \cdot T \cdot \ln(A)}{R_n \cdot R_1 \cdot q} \quad (3.1)$$

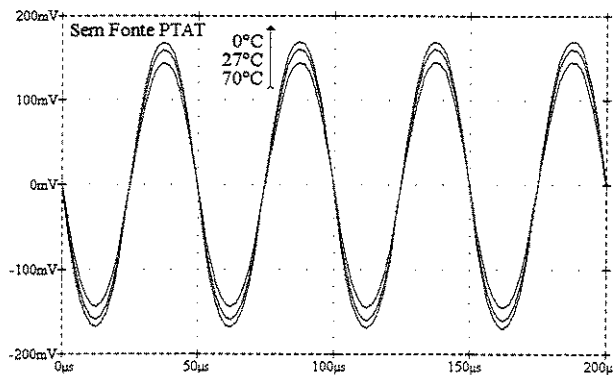


Figura 3.11 (a) – Sem Fonte PTAT

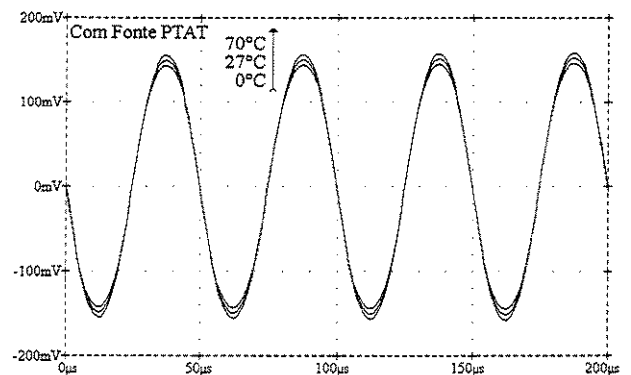


Figura 3.11 (b) – Com Fonte PTAT

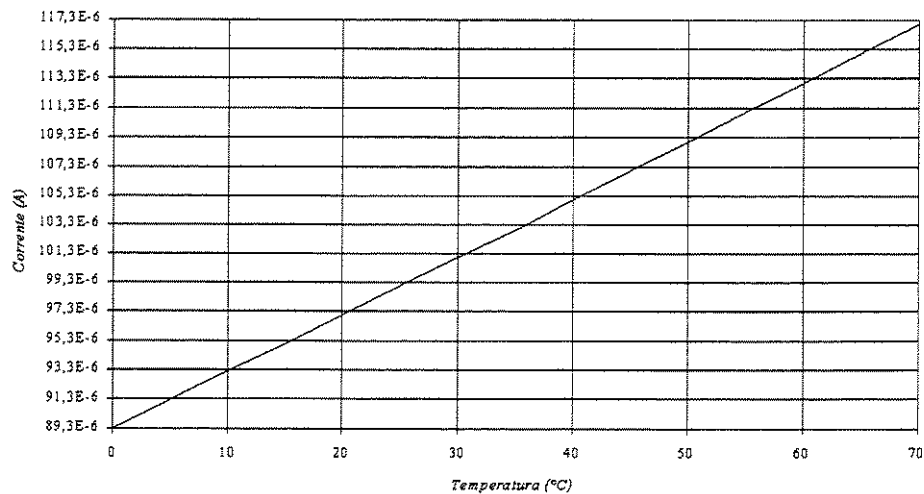


Figura 3.12 – Variação de  $I_{\text{folding}}$  com a Temperatura

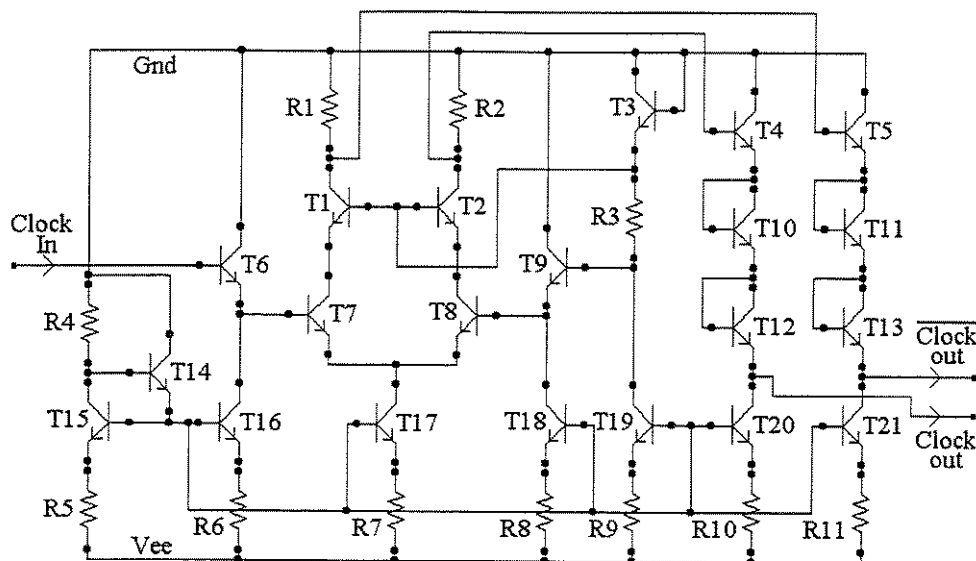


Figura 3.13 – Circuito de Entrada do Sinal de “Clock”

### 3.5 – Circuito de “Clock”:

O sinal de “clock”, em uma fase, é externo ao conversor A/D e deve ter os níveis ECL. Existe um circuito para a entrada do sinal externo de “clock” que trabalha como “buffer”, gera os sinais complementares (“clock” e  $\overline{\text{clock}}$ ) e converte o nível ECL do sinal de entrada para os níveis de voltagem utilizados internamente no conversor A/D. A Fig. 3.13 mostra o circuito utilizado para a entrada do sinal de “clock”. O ideal teria sido utilizar o sinal de entrada de “clock” com duas fases (“clock” e  $\overline{\text{clock}}$ ), porém como todos os geradores de sinais disponíveis para teste (no laboratório) possuem o sinal de saída com apenas uma fase, adotou-se esta opção.

### 3.6 – “Latch” Mestre:

#### 3.6.1 – “Latch” Mestre Simples:

Os sinais  $\overline{\text{MSB}}/\text{MSB}$ ,  $\overline{2\text{SB}}/2\text{SB}$ ,  $\overline{3\text{SB}}/3\text{SB}$ ,  $\overline{\text{S135}}/\text{S135}$ ,  $\overline{\text{S90}}/\text{S90}$ ,  $\overline{\text{S45}}/\text{S45}$ ,  $\overline{\text{SCM}}/\text{SCM}$ ,  $\overline{\text{SC2}}$  e  $\overline{\text{SC2}}$  saem do “folding encoder” e passam por um “latch” mestre simples, cujo circuito pode ser visto na Fig. 3.14. Na saída dos “latches” mestre simples todos os sinais passam a ter um nível digital (veja a Fig. 3.15).

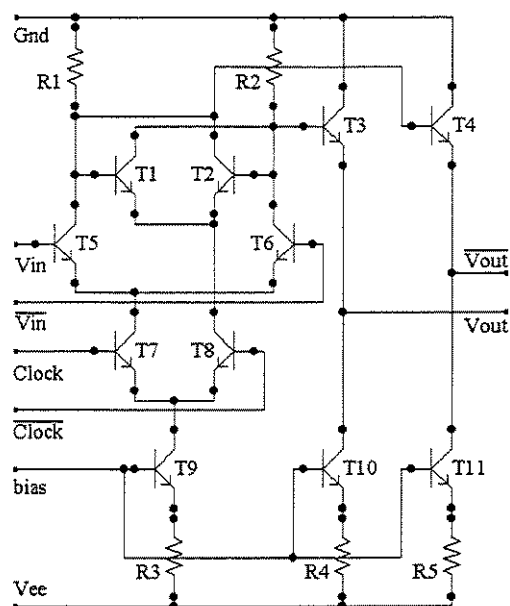


Figura 3.14 – “Latch” Mestre Simples

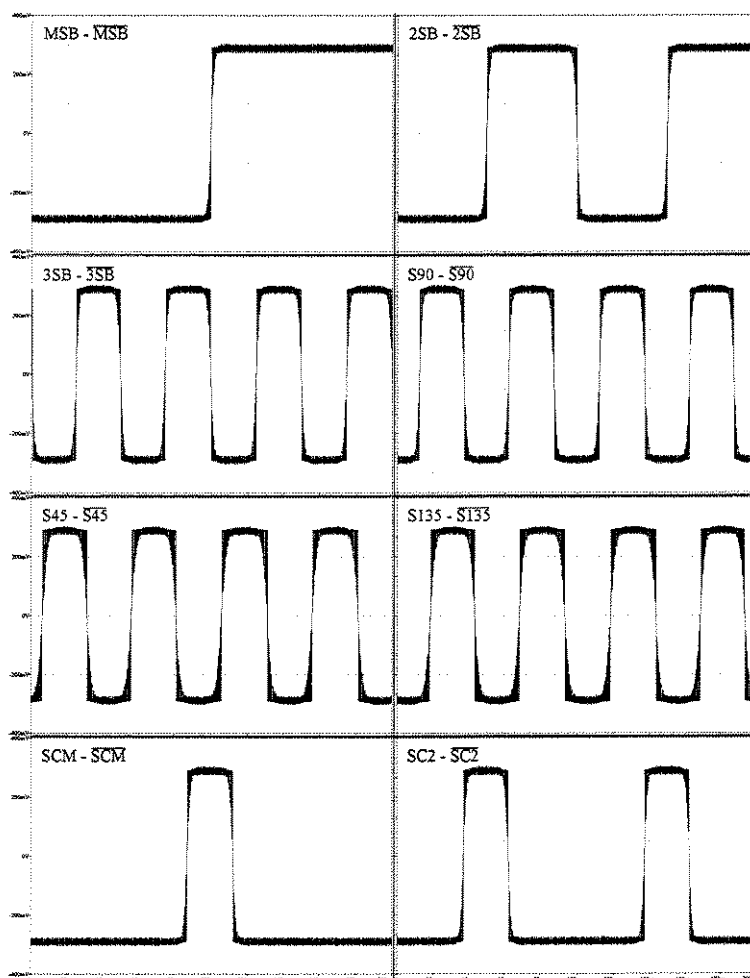


Figura 3.15 – Sinais na Saída dos “Latches” Mestre Simples

### 3.6.2 – “Latch” Mestre Duplo:

Os sinais interpolados são obtidos através da interpolação dupla realizada nos “latches” mestre duplo (o princípio de funcionamento foi visto e analisado no capítulo anterior), cujo circuito pode ser visto na Fig. 3.16. Na Fig. 3.17, os sinais  $3SB/\overline{3SB}$ ,  $S135/\overline{S135}$ ,  $S90/\overline{S90}$  e  $S45/\overline{S45}$  são obtidos no circuito do “folding encoder” e os demais sinais são obtidos por interpolação nos “latches” mestres duplos. Na saída dos “latches” mestre duplos, todos os sinais interpolados têm uma fase (veja a Fig. 3.17) e um nível digital (veja a Fig. 3.18).

O circuito do “latch” mestre duplo (veja a Fig. 3.16) possui dois pré-amplificadores cuja influência relativa no sinal resultante de saída é determinada pela razão das áreas de emissor dos transistores  $T_{11}$  e  $T_9$  (as áreas  $A_2$  e  $A_1$ , respectivamente). A Tabela 3.1 mostra os sinais aplicados nas entradas  $V_{in(a)}/\overline{V_{in(a)}}$  e  $V_{in(b)}/\overline{V_{in(b)}}$ , os respectivos sinais interpolados obtidos na saída e as áreas de emissor utilizadas (normalizadas em relação a referência – veja explicação a seguir). Na prática, as áreas de emissor dos transistores tiveram que ser determinadas com a ajuda dos resultados de simulação, de forma a compensar os efeitos de distorção no sinal e atender às

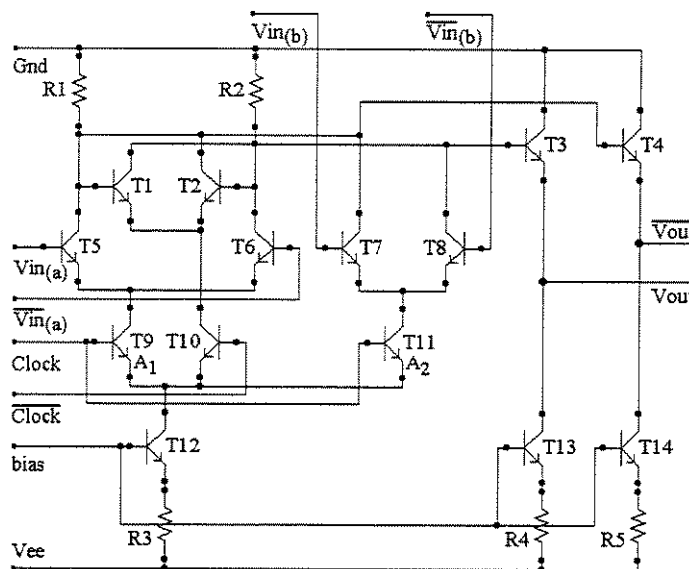


Figura 3.16 – “Latch” Mestre Duplo de Interação

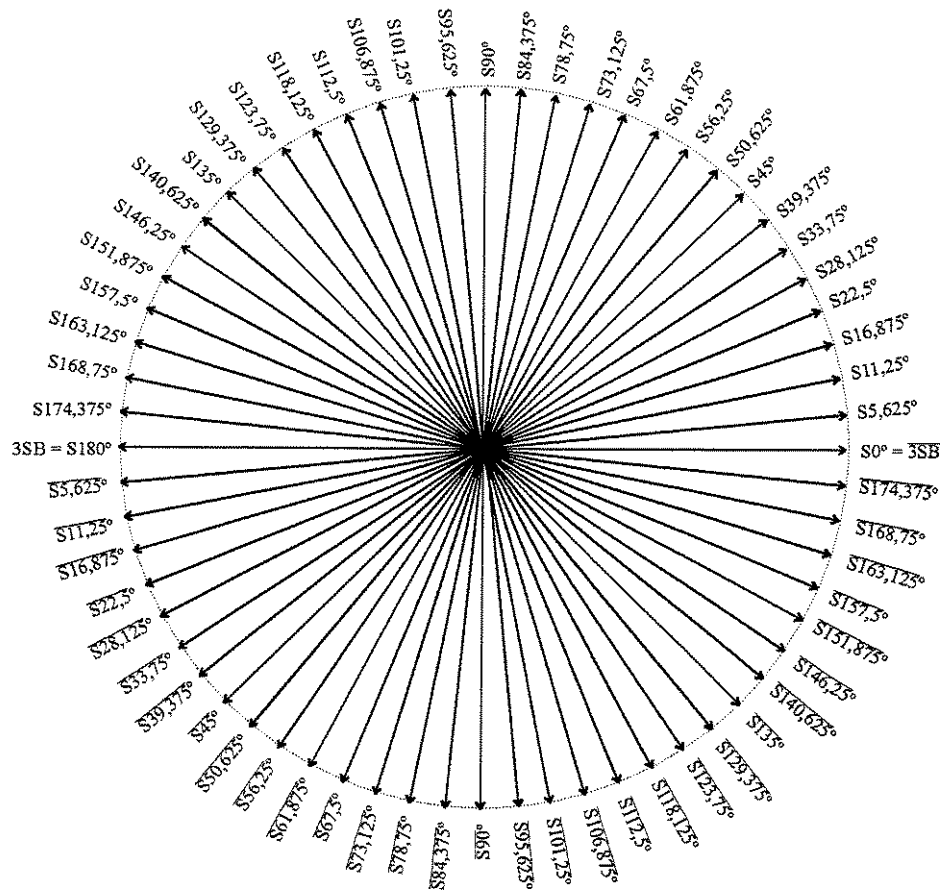


Figura 3.17 – Fase dos Sinais na Saída dos “Latches” Mestre Duplo de Interpolação

condições das regras e parâmetros de projeto do processo utilizado (0,8  $\mu\text{m}$  BiCMOS da Austria Mikro Systeme International AG – AMS). A determinação das áreas considerou uma temperatura de 27°C, as condições típicas para os parâmetros dos modelos de simulação (considerando as tolerâncias para os piores casos) e, ainda, as condições de regras de projeto (para o “layout”), tais como: tamanho mínimo da área do emissor e o “grid” utilizado (de 0,05  $\mu\text{m}$ ), que determinou a máxima precisão possível na determinação das áreas.

No processo utilizado, o emissor tem uma largura fixa igual a 0,8  $\mu\text{m}$ ; por isto, os parâmetros do emissor são referenciados em relação ao seu comprimento de referência igual a 1  $\mu\text{m}$  ( $L_0 = 1 \mu\text{m}$ ). Isto é, nos parâmetros de simulação do processo, o comprimento do emissor “LE” deve ser especificado como a área do componente, sendo “ $\text{AREA} = \text{LE}/L_0$ ”.

Tabela 3.1 – Sinais Aplicados e os Sinais Gerados por Interpolação

| $V_{in(a)}/\sqrt{V_{in(a)}}$ | $V_{in(b)}/\sqrt{V_{in(b)}}$ | Sinal                           | $A_2 = LE/L0$ | $A_1 = LE/L0$ |
|------------------------------|------------------------------|---------------------------------|---------------|---------------|
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S5.625/ $\overline{S5.625}$     | 3,00 $\mu m$  | 6,90 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S11.25/ $\overline{S11.25}$     | 4,60 $\mu m$  | 6,00 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S16.875/ $\overline{S16.875}$   | 5,85 $\mu m$  | 6,00 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S22.5/ $\overline{S22.5}$       | 6,00 $\mu m$  | 5,20 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S28.125/ $\overline{S28.125}$   | 6,00 $\mu m$  | 4,30 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S33.75/ $\overline{S33.75}$     | 5,70 $\mu m$  | 3,00 $\mu m$  |
| S45 / $\overline{S45}$       | $\overline{3SB}$ / 3SB       | S39.375/ $\overline{S39.375}$   | 10,60 $\mu m$ | 3,00 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S50.625/ $\overline{S50.625}$   | 10,60 $\mu m$ | 3,00 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S56.25/ $\overline{S56.25}$     | 5,70 $\mu m$  | 3,00 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S61.875/ $\overline{S61.875}$   | 6,00 $\mu m$  | 4,30 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S67.5/ $\overline{S67.5}$       | 6,00 $\mu m$  | 5,20 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S73.125/ $\overline{S73.125}$   | 5,85 $\mu m$  | 6,00 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S78.75/ $\overline{S78.75}$     | 4,60 $\mu m$  | 6,00 $\mu m$  |
| S45 / $\overline{S45}$       | S90 / $\overline{S90}$       | S84.375/ $\overline{S84.375}$   | 3,00 $\mu m$  | 6,90 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S95.625/ $\overline{S95.625}$   | 3,00 $\mu m$  | 6,90 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S101.25/ $\overline{S101.25}$   | 4,60 $\mu m$  | 6,00 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S106.875/ $\overline{S106.875}$ | 5,85 $\mu m$  | 6,00 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S112.5/ $\overline{S112.5}$     | 6,00 $\mu m$  | 5,20 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S118.125/ $\overline{S118.125}$ | 6,00 $\mu m$  | 4,30 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S123.75/ $\overline{S123.75}$   | 5,70 $\mu m$  | 3,00 $\mu m$  |
| S135 / $\overline{S135}$     | S90 / $\overline{S90}$       | S129.375/ $\overline{S129.375}$ | 10,6 $\mu m$  | 3,00 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S140.625/ $\overline{S140.625}$ | 10,60 $\mu m$ | 3,00 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S146.25/ $\overline{S146.25}$   | 5,70 $\mu m$  | 3,00 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S151.875/ $\overline{S151.875}$ | 6,00 $\mu m$  | 4,30 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S157.5/ $\overline{S157.5}$     | 6,00 $\mu m$  | 5,20 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S163.125/ $\overline{S163.125}$ | 5,85 $\mu m$  | 6,00 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S168.75/ $\overline{S168.75}$   | 4,60 $\mu m$  | 6,00 $\mu m$  |
| S135 / $\overline{S135}$     | 3SB / $\overline{3SB}$       | S174.375/ $\overline{S174.375}$ | 3,00 $\mu m$  | 6,90 $\mu m$  |

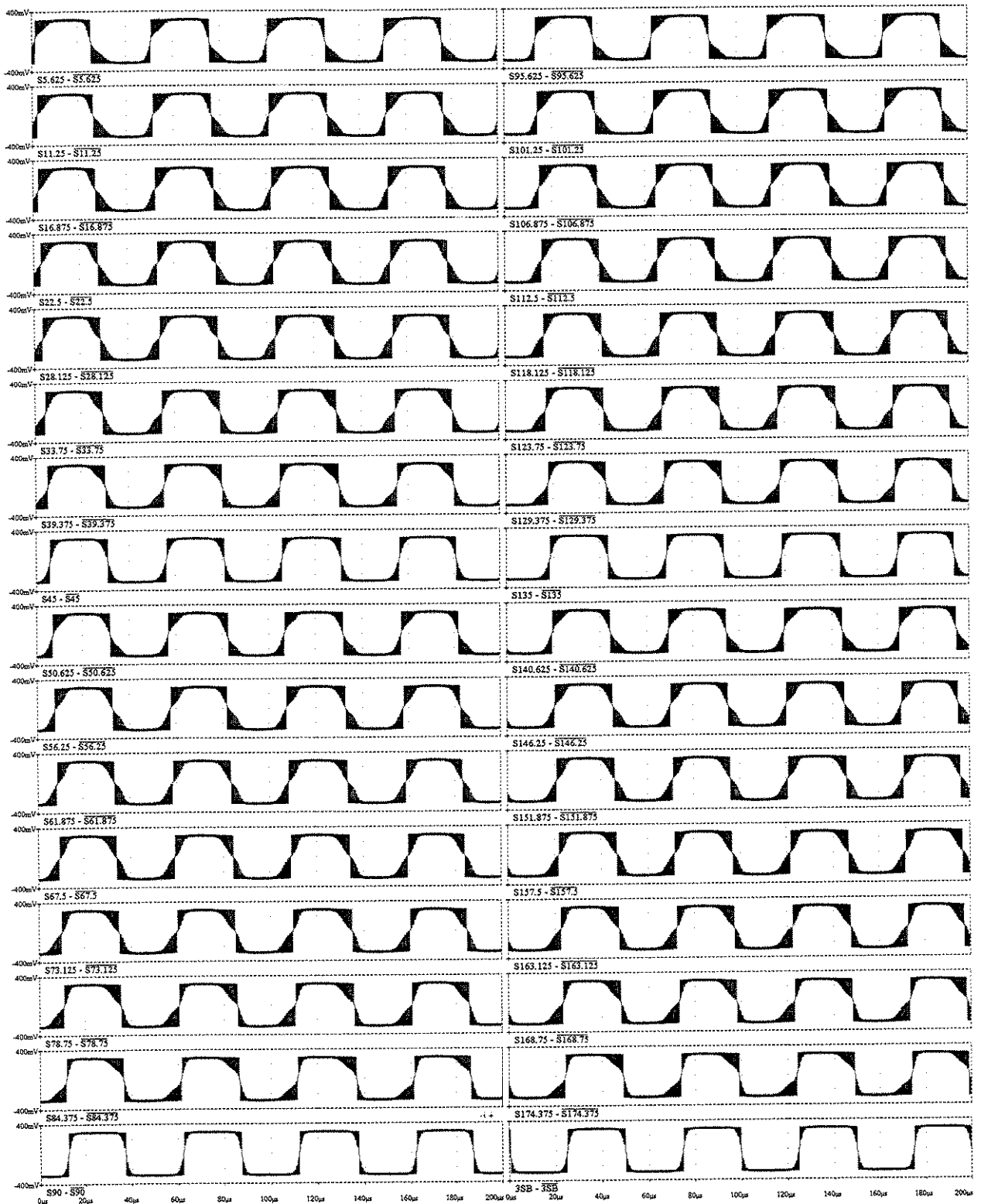


Figura 3.18 – Sinais na Saída dos “Latches” Mestre Duplo de Interpolação

### 3.7 – Portas Ou-Exclusivo com “Latch”:

No sistema “folding” convencional, todos os sinais interpolados, mais os sinais  $3SB/3\overline{SB}$ ,

$S_{135}/\overline{S_{135}}$ ,  $S_{90}/\overline{S_{90}}$  e  $S_{45}/\overline{S_{45}}$  formam um código circular (veja a Tabela 3.2).

Este código poderia ser convertido para uma forma binária usando-se portas “ou-

Tabela 3.2 – Código Circular

[illegible]

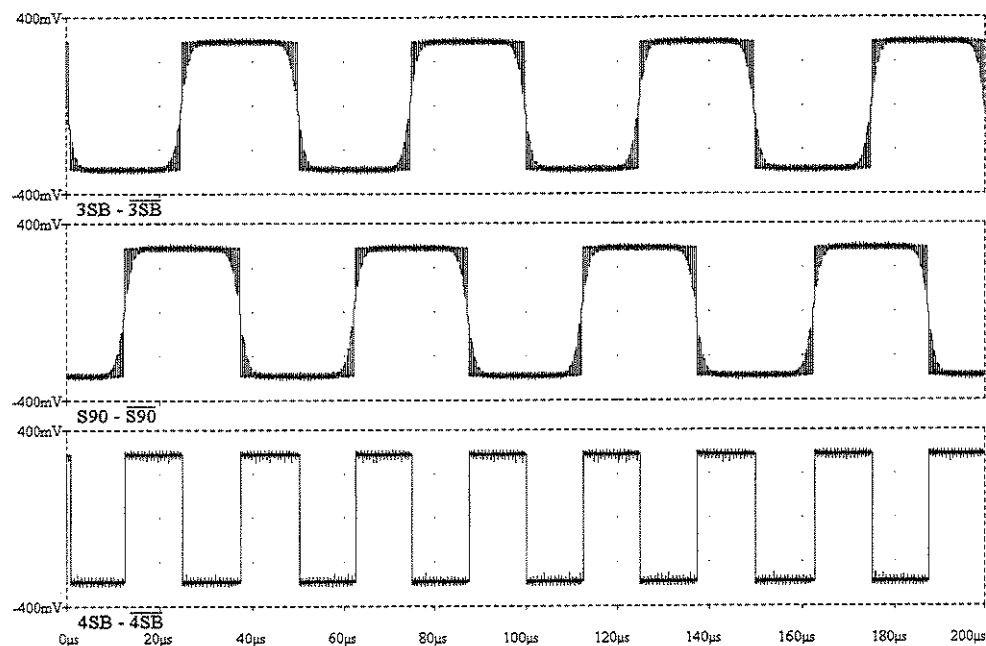


Figura 3.19 – Geração do 4SB

exclusivo” com “latch” e uma estrutura codificadora binária tipo “ROM”. As saídas deste codificador tipo “ROM” seriam os cinco bits menos significativos. Na execução desta operação, seria necessário trinta e duas (32) portas “ou-exclusivo” com “latch” porque esta operação seria realizada no nível dos “latches” escravos do conversor A/D. Na correção dos chamados erros de “bolha” (“bubble error”), seria necessário mais trinta e duas (32) portas “ou-exclusivo” (a técnica aplicada neste conversor A/D para a correção dos erros de bolha será explicada mais adiante neste mesmo capítulo). Nestas duas operações, seria necessário utilizar sessenta e quatro (64) portas “ou-exclusivo” com trinta e dois (32) “latches”. Contudo, é possível reduzir o número de portas “ou-exclusivo” e de “latches” utilizando uma nova topologia para este circuito.

Na Fig. 3.19 é possível ver os sinais  $3SB/\overline{3SB}$  e  $S90/\overline{S90}$ . Se for realizada uma operação “ou-exclusivo” entre estes dois sinais, será obtido o sinal do  $4SB/\overline{4SB}$ . A realização da operação “ou-exclusivo” entre todos os trinta e dois (32) sinais (das Fig. 3.17 e 3.18), que tenham uma diferença de fase de  $90^\circ$ , permitirá a obtenção de uma nova relação de dezesseis (16) sinais

Tabela 3.3 – Sinais Aplicados e os Sinais Resultantes

| Sinal 1                          | Sinal 2                        | Sinal Resultante                         |
|----------------------------------|--------------------------------|------------------------------------------|
| $3SB / \overline{3SB}$           | $S90 / \overline{S90}$         | $4SB / \overline{4SB}$                   |
| $S174.375 / \overline{S174.375}$ | $S84.375 / \overline{S84.375}$ | $168,75^\circ / \overline{168,75^\circ}$ |
| $S168.75 / \overline{S168.75}$   | $S78.75 / \overline{S78.75}$   | $157,5^\circ / \overline{157,5^\circ}$   |
| $S163.125 / \overline{S163.125}$ | $S73.125 / \overline{S73.125}$ | $146,25^\circ / \overline{146,25^\circ}$ |
| $S157.5 / \overline{S157.5}$     | $S67.5 / \overline{S67.5}$     | $135^\circ / \overline{135^\circ}$       |
| $S151.875 / \overline{S151.875}$ | $S61.875 / \overline{S61.875}$ | $123,75^\circ / \overline{123,75^\circ}$ |
| $S146.25 / \overline{S146.25}$   | $S56.25 / \overline{S56.25}$   | $112,5^\circ / \overline{112,5^\circ}$   |
| $S140.625 / \overline{S140.625}$ | $S50.625 / \overline{S50.625}$ | $101,25^\circ / \overline{101,25^\circ}$ |
| $S135 / \overline{S135}$         | $S45 / \overline{S45}$         | $90^\circ / \overline{90^\circ}$         |
| $S129.375 / \overline{S129.375}$ | $S39.375 / \overline{S39.375}$ | $78,75^\circ / \overline{78,75^\circ}$   |
| $S123.75 / \overline{S123.75}$   | $S33.75 / \overline{S33.75}$   | $67,5^\circ / \overline{67,5^\circ}$     |
| $S118.125 / \overline{S118.125}$ | $S28.125 / \overline{S28.125}$ | $56,25^\circ / \overline{56,25^\circ}$   |
| $S112.5 / \overline{S112.5}$     | $S22.5 / \overline{S22.5}$     | $45^\circ / \overline{45^\circ}$         |
| $S106.875 / \overline{S106.875}$ | $S16.875 / \overline{S16.875}$ | $33,75^\circ / \overline{33,75^\circ}$   |
| $S101.25 / \overline{S101.25}$   | $S11.25 / \overline{S11.25}$   | $22,5^\circ / \overline{22,5^\circ}$     |
| $S95.625 / \overline{S95.625}$   | $S5.625 / \overline{S5.625}$   | $11,25^\circ / \overline{11,25^\circ}$   |

resultantes (veja a Tabela 3.3), com diferentes fases em relação ao  $4SB / \overline{4SB}$  (veja a Fig. 3.20).

Desta maneira, será necessário apenas dezesseis (16) portas “ou-exclusivo” com “latch”.

Esta operação é semelhante à que gera o 4SB diretamente com o amplificador “folding” (neste

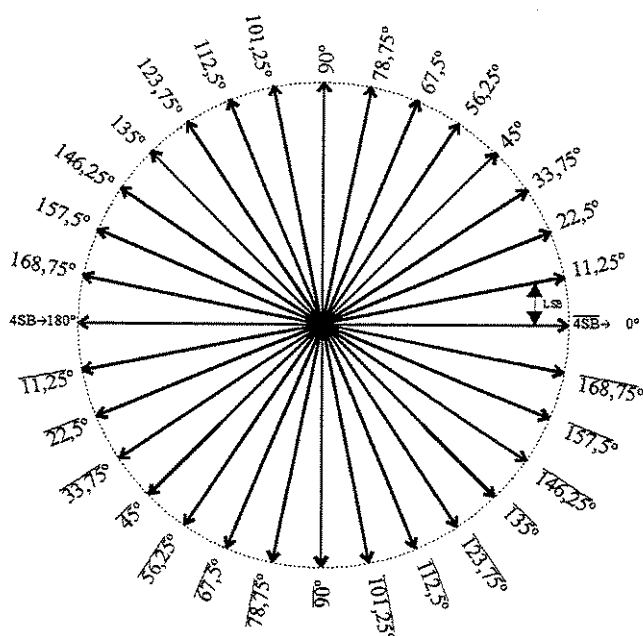


Figura 3.20 – Diagrama de Fase dos Sinais Resultantes

Tabela 3.4 – *Código Circular Resultante*

| 48B | 168.75° | 157.5° | 146.25° | 135° | 123.75° | 112.5° | 101.25° | 90° | 78.75° | 67.5° | 56.25° | 45° | 33.75° | 22.5° | 11.25° |
|-----|---------|--------|---------|------|---------|--------|---------|-----|--------|-------|--------|-----|--------|-------|--------|
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 0      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 0       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 0    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 0      | 0       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 0       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 0   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 1      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 1     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 1      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 1      | 1   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 1      | 1     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 1   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 1       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 1      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 1       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 1       | 1    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 1      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 1       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |
| 1   | 0       | 0      | 0       | 0    | 0       | 0      | 0       | 0   | 0      | 0     | 0      | 0   | 0      | 0     | 0      |

caso, apenas 4 bits seriam gerados por interpolação). Neste caso, o número de sinais necessários se reduz a 16 em vez dos 32 originais. O novo código circular resultante é o mostrado na Tabela 3.4.

Conforme foi dito anteriormente, na topologia convencional, seria necessário sessenta e quatro (64) portas “ou-exclusivo” com trinta e dois (32) “latches”; todos estes circuitos utilizariam aproximadamente novecentos e sessenta (960) transistores e teriam um consumo estimado de cento e noventa miliwatts (190 mW). Na nova topologia, será necessário dezesseis (16) portas “ou-exclusivo” com “latches” e mais trinta e duas (32) portas “ou-exclusivo”; todos estes circuitos utilizarão seiscentos e setenta e dois (672) transistores e terão um consumo estimado de cento e vinte e oito miliwatts (128 mW). A redução no número de transistores é de cerca de trinta por cento (30%) e, no consumo de potência, de cerca de trinta e dois por cento (32%). Estes dados justificam a adoção da nova topologia para o circuito do conversor A/D.

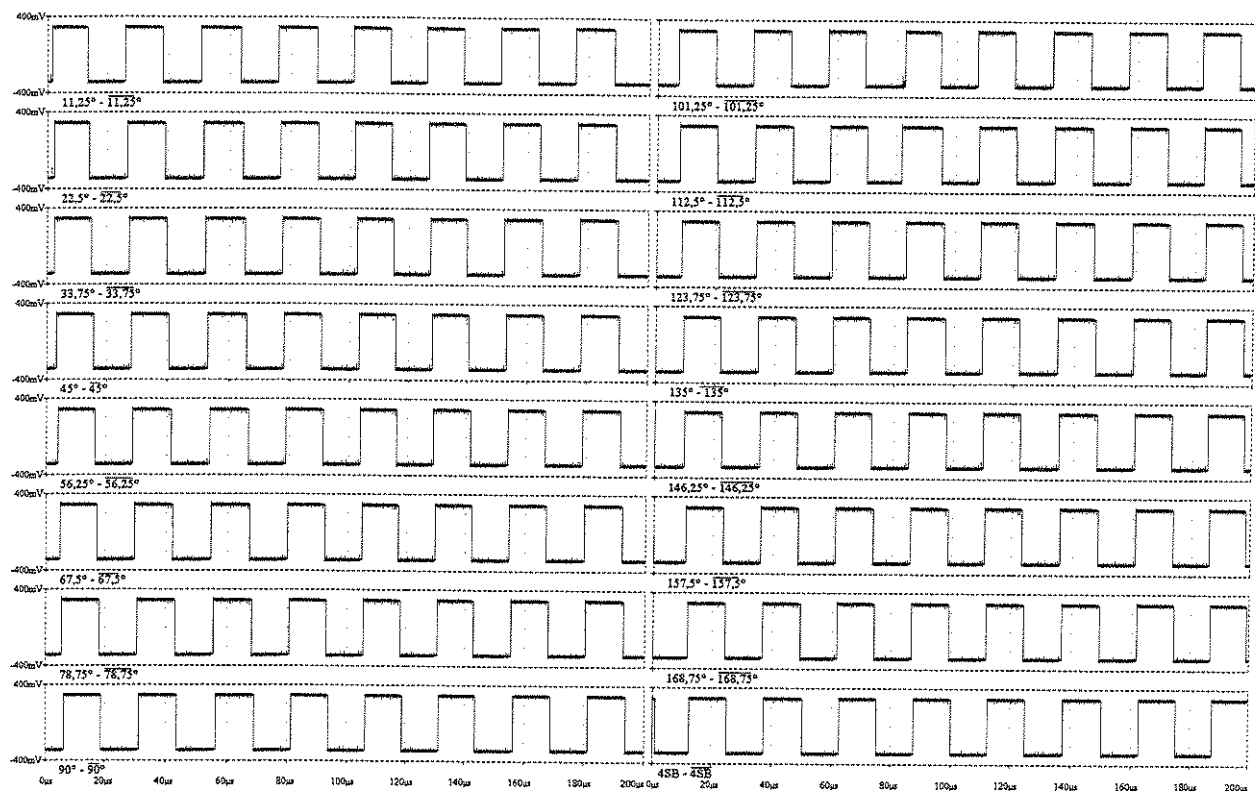


Figura 3.21 – Sinais Resultantes na Saída das Portas “Ou-Exclusivo” com “Latch”

A Fig. 3.21 mostra os dezesseis (16) sinais resultantes gerados e a Fig. 3.22 tem o circuito utilizado na porta lógica “ou-exclusivo” com “latch” (a nível de “latch” escravo).

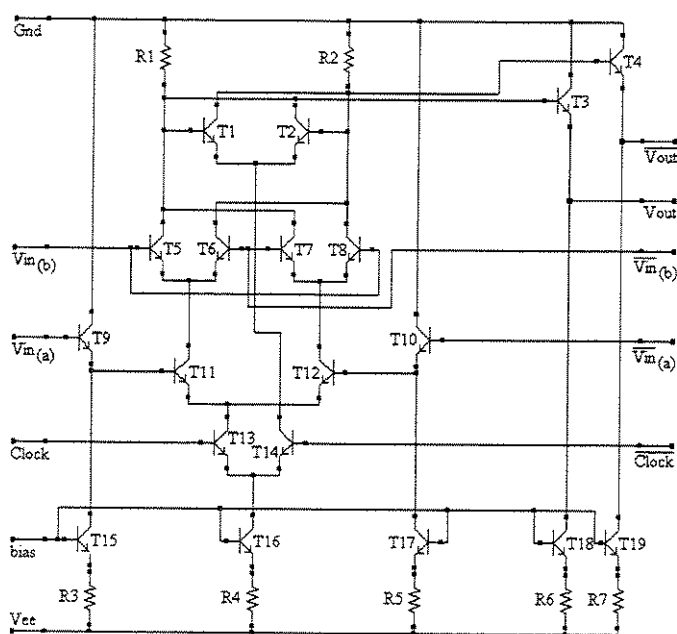


Figura 3.22 – Porta “Ou-Exclusivo” com “Latch”

### 3.8 – Codificação com Correção de Erro:

### 3.8.1 – Codificação:

O novo código circular pode ser convertido em uma forma binária usando-se portas “ou-exclusivo” (EXOR) e uma estrutura codificadora binária tipo “ROM”. As saídas deste codificador “ROM” são os quatro (4) bits menos significativos. Na operação “ou-exclusivo”, realizada entre as saídas adjacentes das portas lógicas “ou-exclusivo” com “latch” (a nível de “latch” escravo - veja a Tabela 3.5), são gerados 16 sinais que podem ser vistos na Fig. 3.23. Nesta figura é apresentado o sinal do  $4SB/4\overline{SB}$  para servir como referência aos sinais de saída das portas “ou-exclusivo”. Estes sinais também foram divididos em dois gráficos para melhor

Tabela 3.5 – Saídas das Porta Ou-Exclusivo

[illegible]

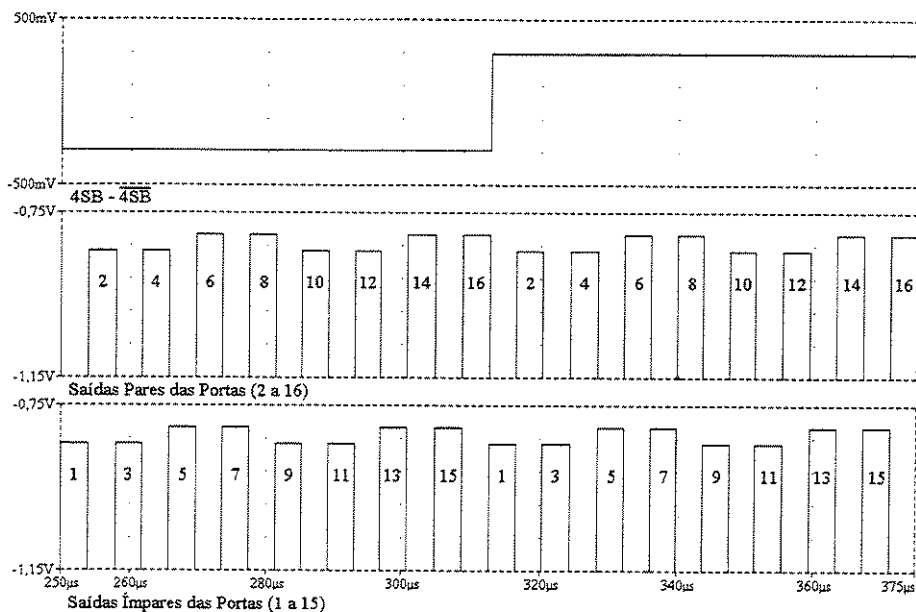


Figura 3.23 – Sinais na Saída das Portas “Ou-Exclusivo”

visualização. É possível verificar que os 16 sinais se repetem num período de meio 4SB, o que corresponde a um 5SB. A correta composição destes sinais irá formar os 4 bits menos significativos do conversor A/D. Apesar das amplitudes destes sinais apresentarem uma pequena diferença, isto não acarreta nenhum problema porque serão corrigidas nas portas lógica de saída.

### 3.8.2 – Correção de Erro de Bolha:

Na decodificação do código circular, é detectada a transição de zero para um binário (usando-se portas “ou-exclusivo” - EXOR) e os circuitos irão endereçar uma linha da estrutura codificadora binária tipo “ROM”. Se existir algum erro de bolha, haverá pelo menos dois pontos de transição de zero para um no código e os circuitos irão endereçar duas linhas da estrutura codificadora binária tipo “ROM”. Endereçamento simultâneo de duas linhas da estrutura codificadora binária tipo “ROM” irá causar uma descontinuidade, ou “glitch”, na forma de onda reconstruída porque irá ocorrer nível lógico inválido.

No processo de votação para correção de erro de bolha, cada saída da porta “ou-exclusivo” com “latch” (a nível de “latch” escravo e gerador do código circular resultante com 16

sinais, veja o item 3.7) seria examinada relativamente aos seus dois vizinhos próximos, e a saída seria trocada se ela discordasse de ambos. A equação lógica para a saída corrigida seria:

$$C_n = (C_n \text{ AND } C_{n-1}) \text{ OR } (C_n \text{ AND } C_{n+1}) \text{ OR } (C_{n-1} \text{ AND } C_{n+1}) \quad (3.2)$$

Onde:

$C_n$  é a saída da porta “ou-exclusivo” com “latch” (n).

Nesta equação, a saída corrigida seria o estado majoritário de três saídas adjacentes das portas “ou-exclusivo” com “latch”. Após esta correção, o ponto de transição de zero para um deveria ser identificado para endereçar a linha da “ROM”. Isto é, nós teríamos uma função “ou-exclusivo” (EXOR).

$$R\{n\} = (C_n \text{ AND not } C_{n+1}) \text{ OR } (\text{not } C_n \text{ AND } C_{n+1}) \quad (3.3)$$

Onde:

$R\{n\}$  é a linha endereçada da “ROM”.

$C_n$  é a saída da porta “ou-exclusivo” com “latch” (n).

Na prática, os dois passos de correção e identificação do ponto de transição de zero para um foram combinados em uma única operação. Conceitualmente, esta operação pode ser caracterizada pela equação abaixo. Esta equação conduz a um circuito lógico muito simples mostrado na Fig. 3.24 e possibilita a implementação de um “hardware” muito eficiente. O circuito lógico de correção incrementa somente uma porta “EXOR” porque a função “AND” pode ser feita por lógica “and-wired”. A implementação do circuito pode ser vista na Fig. 3.25.

$$R\{n\} = (C_{n-2} \text{ EXOR } C_{n+1}) \text{ AND } (C_{n-1} \text{ EXOR } C_n) \quad (3.4)$$

Onde:

$R\{n\}$  é a linha endereçada da “ROM”.

$C_n$  é a saída da porta “ou-exclusivo” com “latch” (n).

De fato, o resultado indica que a proposta de correção de erro está baseada na análise de

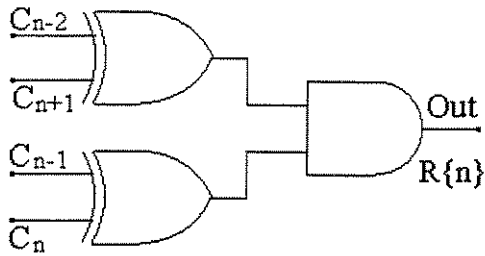


Figura 3.24 – Circuito Lógico de Correção

um conjunto de quatro saídas das portas “ou-exclusivo” com “latch”. Elas são vizinhas próximas e têm um código circular de saída. Isto é, o ponto de transição de zero para um deve ser

identificado e corrigido com o exame relativo da saída de quatro portas “ou-exclusivo” com “latch”. Este esquema de correção de erro tem um processo de votação e a saída da porta “ou-exclusivo” com “latch” é examinada em relação às saídas dos seus três vizinhos próximos e a saída é corrigida se elas discordarem. Assim, os dois passos de correção e identificação do ponto de transição de zero para um são combinados em um única operação lógica. Na prática, o esquema de correção de erro é feito durante a detecção da transição de zero para um do código circular. O ponto de transição de zero para um é identificado e corrigido e será endereçada somente uma linha da estrutura tipo “ROM”.

### 3.8.3 – Resultados e Limitações:

A maior probabilidade de ocorrência de padrão de erro de bolha é a do erro 1, seguido pelo erro 2 e erro 3 da Tabela 1.4 no primeiro capítulo [22]. Para exemplo do comportamento do

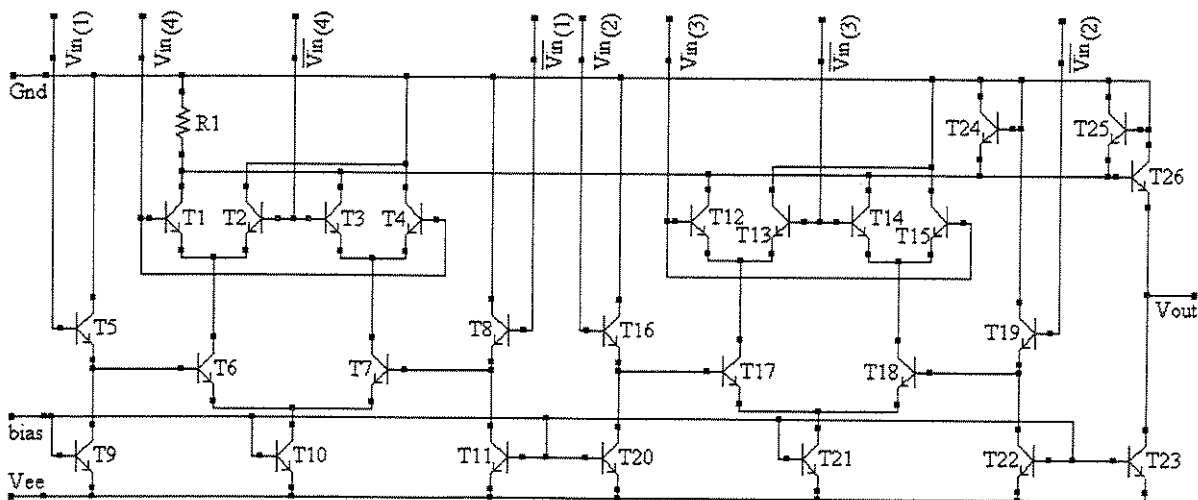


Figura 3.25 – Circuito de Correção

Tabela 3.6 – Exemplos de Correção de Erro

| Erro   | Erro de Bolha     | Correção         | Veja a Fig. |
|--------|-------------------|------------------|-------------|
| Erro 1 | 00000000000001011 | 0000000000000111 | 3.26        |
| Erro 2 | 0000000001101111  | 0000000001111111 | 3.26        |
| Erro 3 | 0001001111111111  | 0000001111111111 | 3.26        |

método proposto, foi feita uma análise dos 3 erros básicos para o código circular resultante de 16 “bits”. Esta proposta de correção de erro torna possível corrigir os erros 1, 2 e 3 (veja a Tabela 3.6) mas irá falhar no caso extremo de um ocasional erro 4.

Para o erro de bolha 1, a melhor correção é “0000000000000111” [7, 18 e 22] e um exemplo da forma de onda reconstruída pode ser visto na Fig. 3.26. Nesta figura, quando o erro de bolha ocorre, a forma de onda não corrigida tem um código indefinido e a forma de onda corrigida mantém o código anterior.

Para o erro de bolha 2, a melhor suposição é “0000000001111111” e um exemplo da forma de onda reconstruída também pode ser visto na Fig. 3.26. Da mesma forma, quando o erro

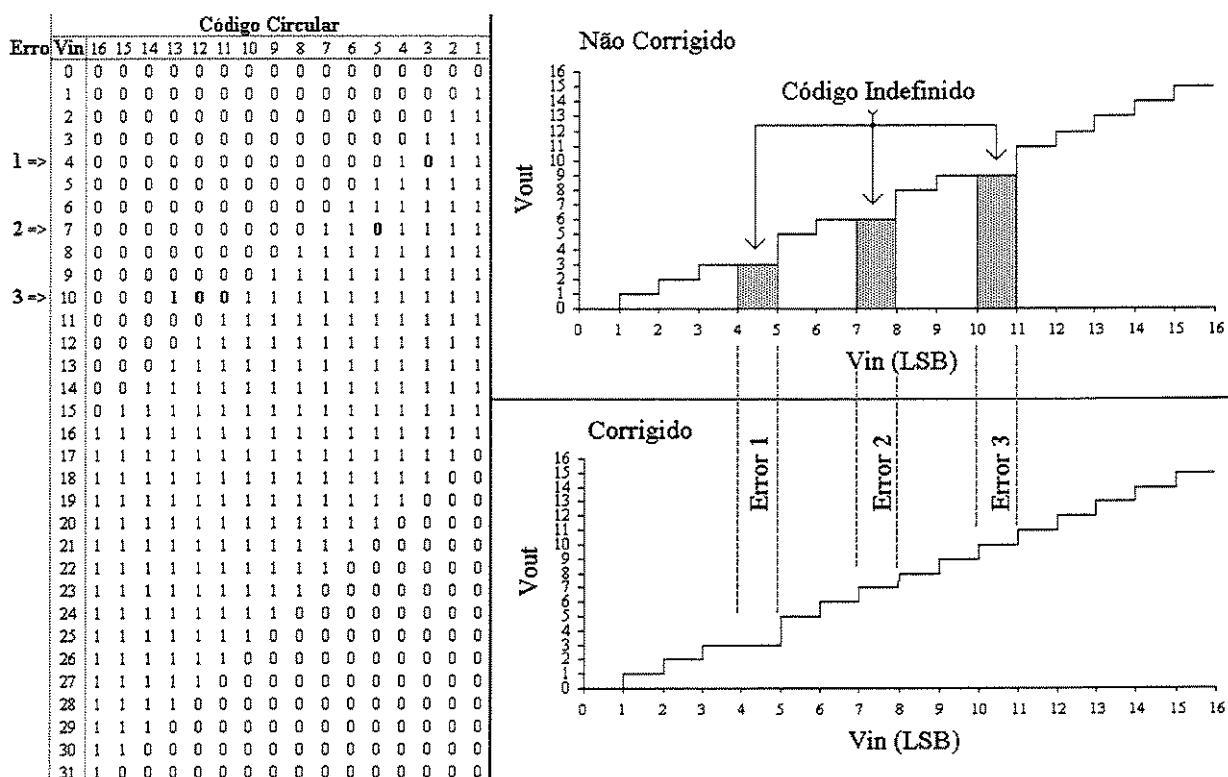


Figura 3.26 – Código Circular e o Circuito de Correção

de bolha ocorre, a forma de onda não corrigida tem um código indefinido e a forma de onda corrigida recupera o valor correto do código.

Para o erro de bolha 3, existem duas possibilidades [22]. A primeira possibilidade é a de que o “1” isolado esteja errado; neste caso, a melhor suposição resultaria em “0000001111111111”. A segunda possibilidade é a de que cada “bit” errado é igualmente provável de estar errado, com uma chance de 33% de ser verdadeiro. Neste caso, existem três opções. A melhor suposição seria uma das “0000001111111111” ou “0000011111111111” ou “0000111111111111”. Para este método proposto de correção de erro, o erro de bolha é corrigido para “0000001111111111”, que é uma das melhores suposições e é uma opção de correção que atende às duas possibilidades. Na Fig. 3.26, é possível ver que a forma de onda não corrigida teve um código indefinido e a forma de onda corrigida recuperou o valor correto do código.

Estes três padrões de correção de erro de bolha são os mesmos propostos em [07, 08 e 18] para os erros 1, 2 e 3, respectivamente.

#### 3.8.4 – Segunda Correção de Erro:

A informação oriunda dos “bits” menos significativos é usada para corrigir qualquer erro de temporização nos “bits” mais significativos antes deles serem aplicados nas portas de saída “ECL”. Se os MSB’s estão na região de transição, eles são substituídos por um valor apropriado baseado no “bit” mais significativo dos LSB’s. Isto é, os MSB’s são substituídos pelo mais significativo “bit” dos LSB’s ou pelo seu complemento dependendo da região de transição. Neste caso, o “bit” mais significativo dos LSB’s é chamado de “extra-bit” e os MSB’s são sincronizados com o resto do sistema usando-se o “extra-bit” (veja a Fig. 3.1). É necessário uma correção adicional no “extra-bit” porque ele não é corrigido na correção anterior. O esquema da correção de erro anterior é feito durante a detecção da transição de zero para um do código circular e o “extra-bit” não é corrigido. Deste modo, é necessário uma correção específica (que

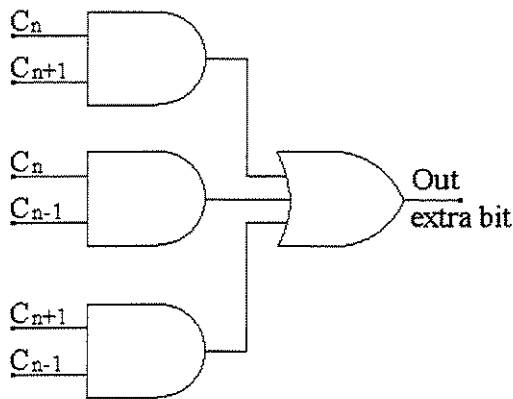


Figura 3.27 – Correção do “Extra Bit”

caracterizada pela equação abaixo.

$$C_n = (C_n \text{ AND } C_{n+1}) \text{ OR } (C_n \text{ AND not } C_{n-1}) \text{ OR } (C_{n+1} \text{ AND not } C_{n-1}) \quad (3.5)$$

Onde:

$C_n$  é a saída da porta “ou-exclusivo” com “latch” (n).

Na implementação do circuito da Fig. 3.27, para as portas AND, foi utilizado o circuito da Fig. 3.28 e para a porta OR foi utilizado o circuito da Fig. 3.29 (que é uma porta AND de três entradas, utilizada como NAND com as entradas invertidas  $\rightarrow$  OR).

### 3.8.5 – Análise da Correção de Erro:

De fato, seria possível corrigir todos os erros de código, mas isto resultaria em um circuito muito grande de correção. O objetivo foi implementar um circuito com pequena complexidade, simples implementação e que corrija a maioria dos erros de código. Deveria existir uma porta “ou-exclusivo” entre cada par de portas “ou-exclusivo” com “latch” que fossem adjacentes. Se nós usarmos um circuito lógico “ou-exclusivo” com correção, cada um destes circuitos lógicos terá incrementado somente uma porta EXOR porque a função AND poderá ser feita por lógica “and-wired”. Deste modo, é possível implementar um circuito lógico que tenha pequena área e baixo consumo. A segunda correção não incrementa significativamente a área total do circuito porque é um único circuito que não se repete.

foi chamada de segunda correção de erro) para o “extra-bit”. Este esquema de correção de erro também tem um processo de votação e a saída do “extra-bit” é examinada em relação às saídas dos seus dois vizinhos próximos e é corrigida se discordarem.

Conceitualmente, esta operação pode ser

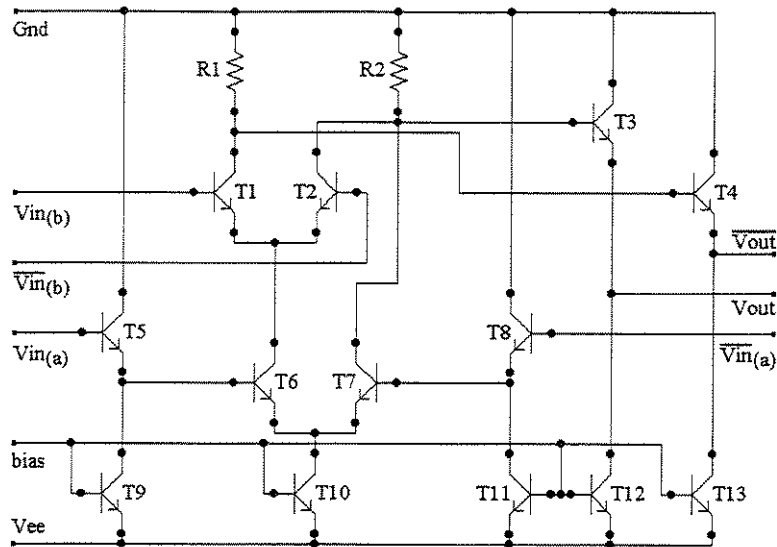


Figura 3.28 – Porta AND com Duas Entradas

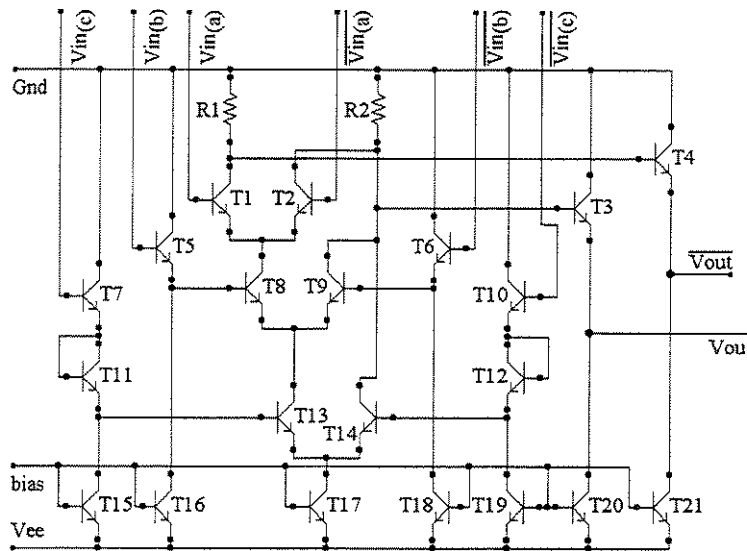


Figura 3.29 – Porta AND com Três Entradas

Realmente, a técnica de correção de erro, a qual detecta e corrige erros de bolha durante a detecção da transição de zero para um do código circular, melhora a razão de erro do conversor A/D. Quando a razão de erro é extremamente alta, o sinal digitalizado aparece distorcido e a razão sinal/ruído deteriora rapidamente. Para muitas aplicações, é necessário uma baixa razão de erro. Assim, a correção de erros de bolha também melhora a performance em alta frequência do conversor A/D [18]. Isto é a principal evidência para o sucesso dos circuitos propostos de correção de erro.

### 3.8.6 – Estrutura Tipo “ROM”:

A estrutura tipo “ROM” combina os sinais de saída das portas “ou-exclusivo” com correção de erro em 8 sinais complementares (veja a Fig. 3.30) [06 e 08]. Estes sinais são: 5SB,  $\overline{5SB}$ , 6SB,  $\overline{6SB}$ , 7SB,  $\overline{7SB}$ , LSB e  $\overline{LSB}$ . Veja, a seguir, as combinações dos sinais de saída das portas “ou-exclusivo” com correção de erro na estrutura tipo “ROM” e compare a Tabela 3.5 com as Fig. 3.30 e 3.31:

$$5SB \Rightarrow (16 + 15 + 14 + 13 + 12 + 11 + 10 + 9)$$

$$\overline{5SB} \Rightarrow (8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)$$

$$6SB \Rightarrow (16 + 15 + 14 + 13 + 8 + 7 + 6 + 5)$$

$$\overline{6SB} \Rightarrow (12 + 11 + 10 + 9 + 4 + 3 + 2 + 1)$$

$$7SB \Rightarrow (16 + 15 + 12 + 11 + 8 + 7 + 4 + 3)$$

$$\overline{7SB} \Rightarrow (14 + 13 + 10 + 9 + 6 + 5 + 2 + 1)$$

$$LSB \Rightarrow (16 + 14 + 12 + 10 + 8 + 6 + 4 + 2)$$

$$\overline{LSB} \Rightarrow (15 + 13 + 11 + 9 + 7 + 5 + 3 + 1)$$

## 3.9 – Sincronismo de Bits (“Bit Sync”):

### 3.9.1 – Sinais de Controle:

A solução é detectar os instantes de transição dos bits mais significativos e, no momento

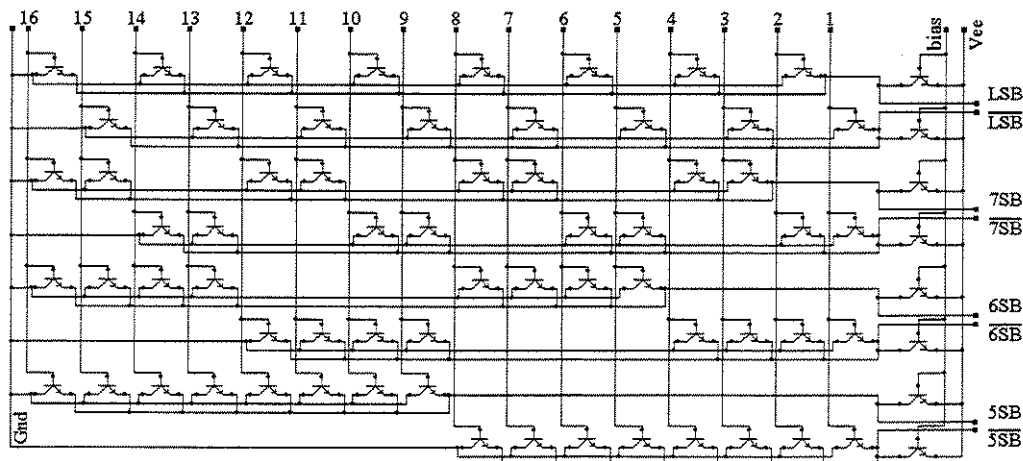


Figura 3.30 – Estrutura Tipo ROM

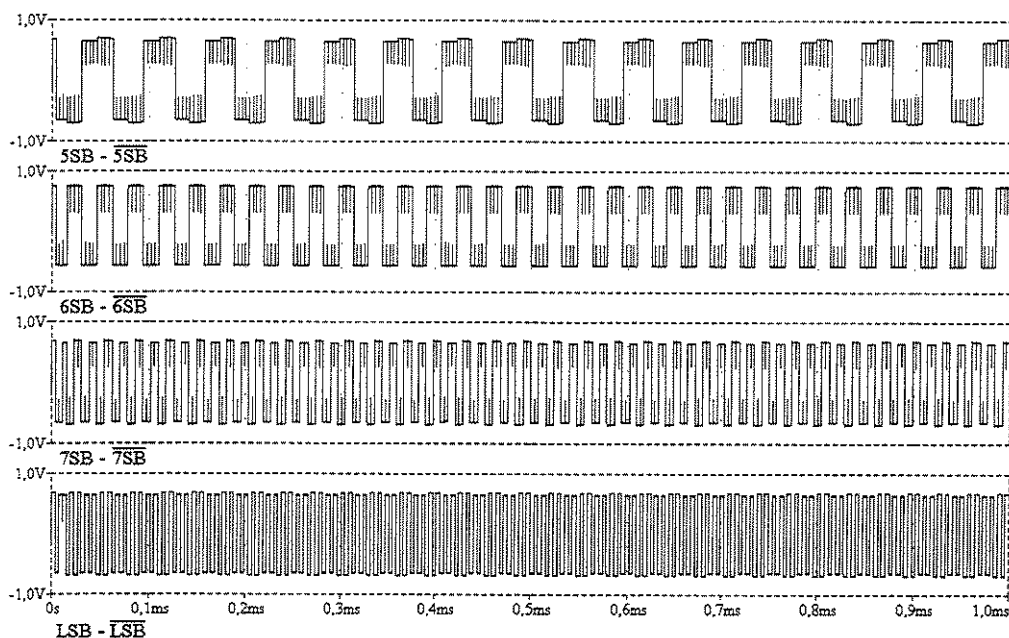


Figura 3.31 – Sinais de Saída da Estrutura Tipo ROM

adequado, substituí-los por um valor apropriado. Em função da nova topologia aplicada neste conversor A/D, utilizou-se o quarto bit como referência porque ele também participa da geração dos bits menos significativos através da interpolação. A sincronização dos bits demonstrada no item 1.8 do capítulo 1 era mais fácil de ser elaborada porque apenas o MSB e o 2SB eram sincronizados com o 3SB (que era o “extra bit”). Nesta sincronização proposta, o MSB, 2SB e 3SB devem ser sincronizados com o 4SB (que agora é o “extra bit”).

Isto é, a informação oriunda do quarto bit é usada para corrigir qualquer erro de

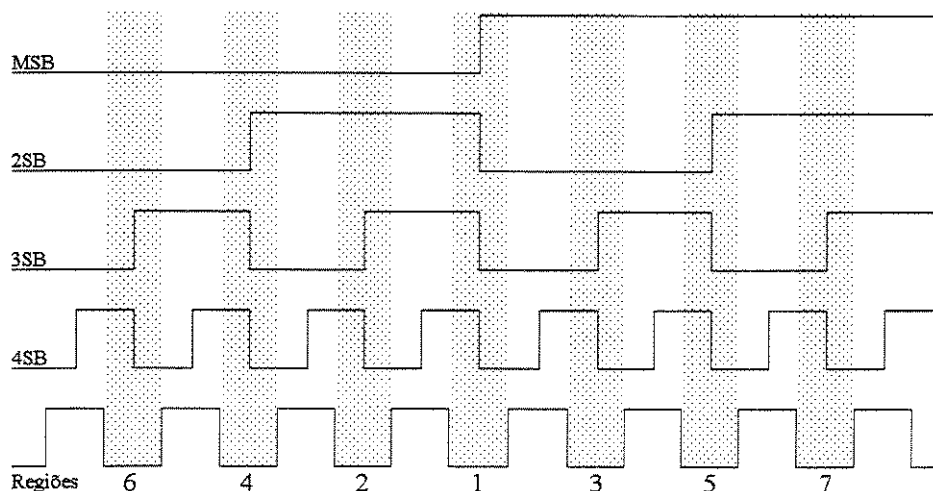


Figura 3.32 – Regiões de Transição

Tabela 3.7 – *Correção de Erro de Sincronismo*

| Bit/Região | “1”              | “2”              | “3”              | “4”              | “5” | “6”              | “7”              |
|------------|------------------|------------------|------------------|------------------|-----|------------------|------------------|
| MSB        | $\overline{4SB}$ | -                | -                | -                | -   | -                | -                |
| 2SB        | 4SB              |                  |                  | $\overline{4SB}$ | 4SB |                  |                  |
| 3SB        | 4SB              | $\overline{4SB}$ | $\overline{4SB}$ | 4SB              | 4SB | $\overline{4SB}$ | $\overline{4SB}$ |

temporização nos bits mais significativos (MSB, 2SB e 3SB) antes deles serem aplicados nas portas de saída “ECL”. Se os bits mais significativos estão na região de transição, eles são substituídos por um valor apropriado baseado no quarto bit. Portanto, os bits mais significativos são substituídos pelo quarto bit ou pelo seu complemento dependendo da região de transição. Neste caso, o quarto bit, depois de corrigido (veja o item 3.8.4 neste capítulo), é chamado de “extra-bit” e os bits mais significativos são sincronizados com o resto do sistema utilizando-se o “extra-bit”.

Na Fig. 3.32 é possível ver as sete regiões de transição que podem ser facilmente identificadas. Na região “1”, no instante de transição, o sinal do MSB deve ser substituído pelo  $\overline{4SB}$ , o 2SB deve ser substituído pelo 4SB e o 3SB deve ser substituído pelo 4SB. Nas regiões “2”, “3”, “6” e “7” não existe transição do MSB e do 2SB; porém, o 3SB deve ser substituído pelo  $\overline{4SB}$ . Nas regiões “4” e “6” não existe transição do MSB; o 2SB deve ser substituído pelo  $\overline{4SB}$  e o 3SB deve ser substituído pelo 4SB. Todas combinações possíveis podem ser vistas na Tabela 3.7. Para fazer todas estas substituições, é necessário um circuito lógico extra para identificar o exato momento em que cada região de transição ocorre e habilitar os circuitos lógicos de sincronismo. Neste instante, os circuitos lógicos de sincronismo atuam e fazem a substituição necessária dos três (3) bits mais significativos pelo quarto bit (“extra bit”).

A Fig. 3.33 mostra o circuito lógico utilizado na obtenção dos cinco (5) sinais de controle (“A”, “B”, “C”, “AB” e “ABC”) necessários para operarem as substituições dos bits mais significativos pelo quarto bit (“extra bit”) e na Fig. 3.34 são mostradas as formas de onda

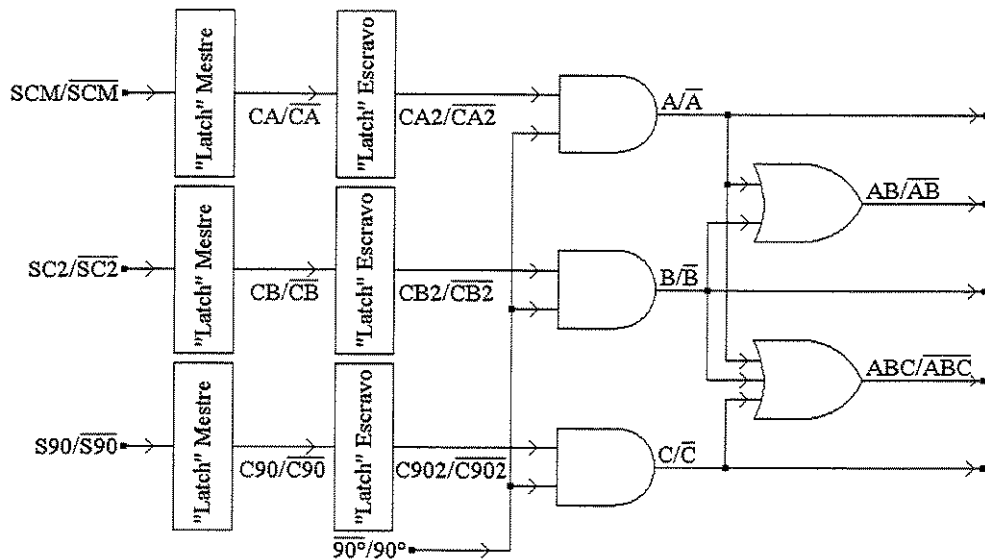


Figura 3.33 – Circuito Lógico para Obtenção dos Sinais de Controle

(internas e de saída) do circuito lógico para obtenção dos sinais de controle.

Os sinais  $S90/\overline{S90}$ ,  $SC2/\overline{SC2}$  e  $SCM/\overline{SCM}$  são gerados no circuito do “folding encoder” e passam pelos “latches” mestres, que têm na saída os sinais  $C90/\overline{C90}$ ,  $CB/\overline{CB}$  e  $CA/\overline{CA}$ . Posteriormente, os sinais passam pelos “latches” escravos, que têm na saídas os sinais  $C902/\overline{C902}$ ,  $CB2/\overline{CB2}$  e  $CA2/\overline{CA2}$ . Contudo, estes sinais ficam em nível alto durante  $3SB/2$  (meio 3SB), o que não satisfaz porque eles precisam ficar em nível alto durante apenas  $4SB/2$  (meio 4SB). Para obter esta correção, é realizada uma operação “AND” com o sinal  $90^\circ/90^\circ$  (o mesmo da Fig. 3.20 e Fig. 3.21). A Fig. 3.34 mostra todos estes sinais e ainda o sinal  $4SB/4SB$  para comparação. Neste ponto já existem os sinais “A”, “B” e “C”. A seguir, são feitas duas somas para obtenção dos sinais “AB” e “ABC”. Desta forma, são obtidos os cinco (5) sinais de controle necessários. Os circuitos utilizados para o “latch” mestre e para o “latch” escravo é o da Fig. 3.14. O Circuito utilizado nas portas “AND” e na porta “OR” de duas (2) entradas (operação feita por uma porta “NAND”) é o da Fig. 3.28 e o circuito utilizado na porta “OR” de três (3) entradas (operação também feita por uma porta “NAND”) é o da Fig. 3.29.

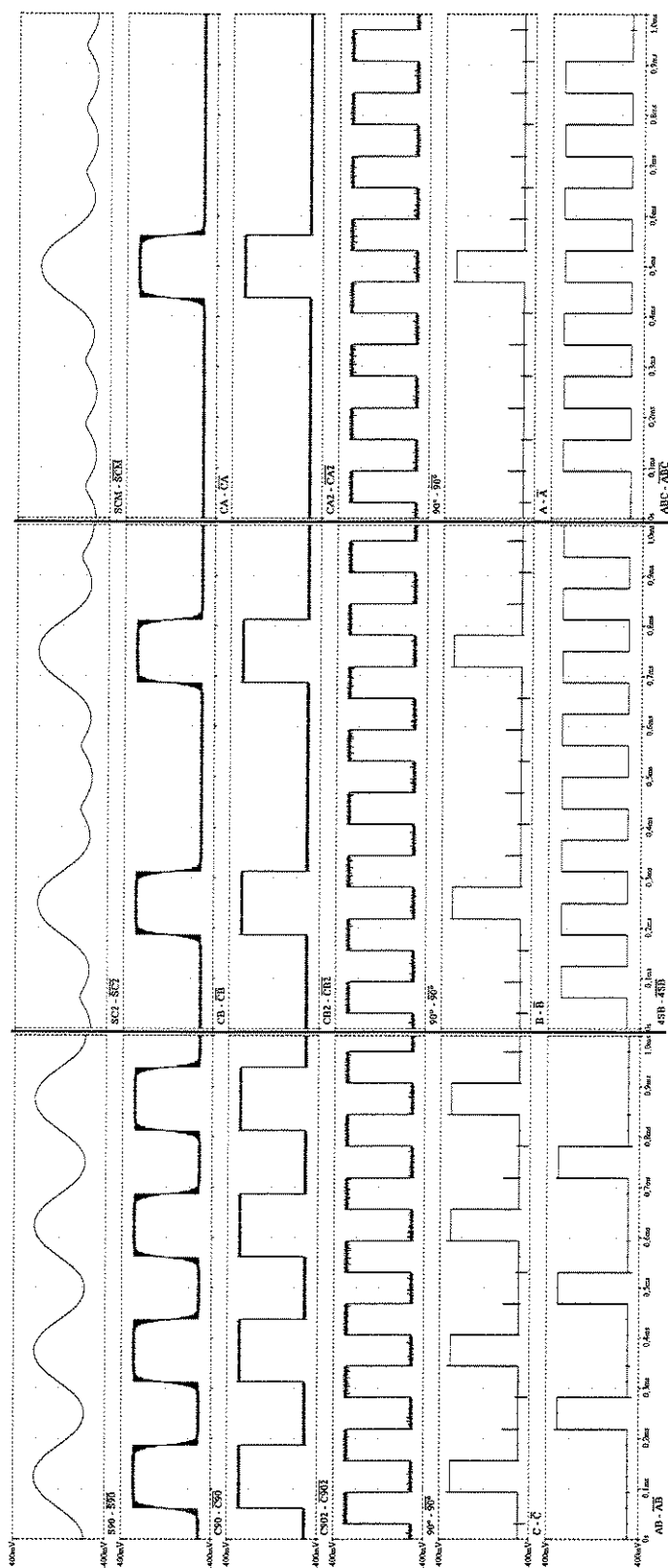


Figura 3.34 – Sinais de Controle

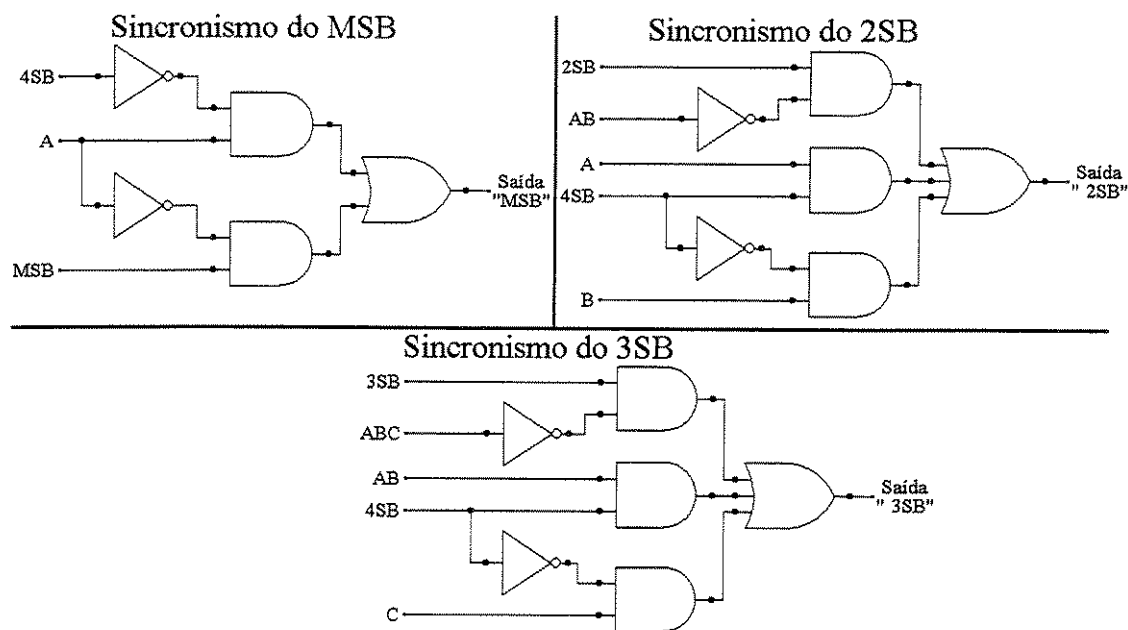


Figura 3.35 – Circuito Lógico de Correção do MSB, 2SB e 3SB

### 3.9.2 – Sincronização dos Bits:

A Fig. 3.35 mostra os três circuitos lógicos que fazem a correção de erro de sincronismo

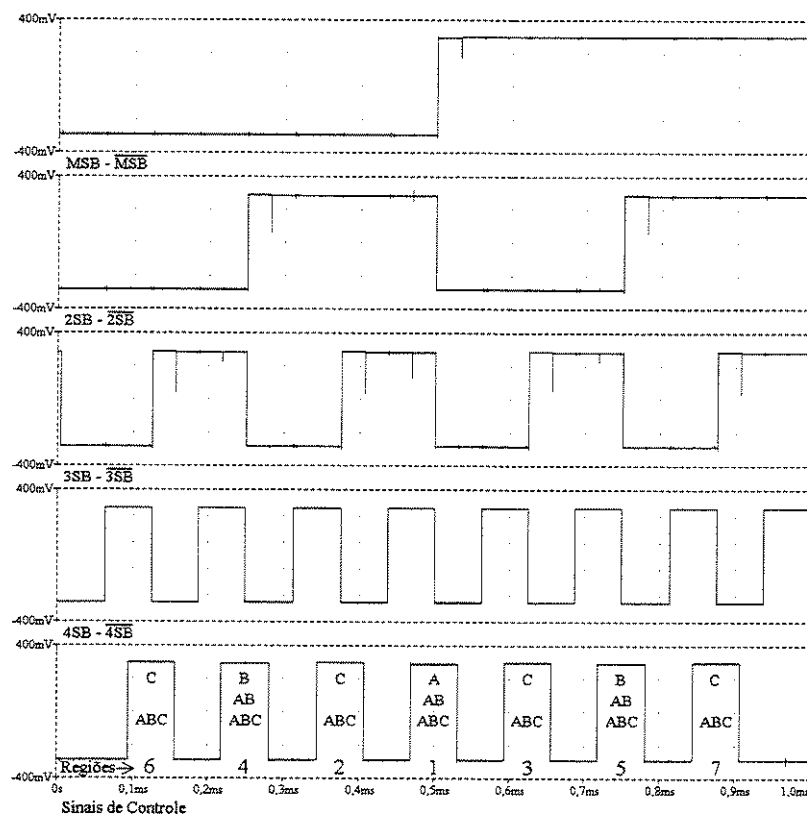


Figura 3.36 – Sinais  $MSB/\overline{MSB}$ ,  $2SB/\overline{2SB}$ ,  $3SB/\overline{3SB}$ ,  $4SB/\overline{4SB}$  e os Sinais de Controle

de bit e a Fig. 3.36 mostra os sinais  $MSB/\overline{MSB}$ ,  $2SB/\overline{2SB}$ ,  $3SB/\overline{3SB}$ ,  $4SB/\overline{4SB}$  e os cinco (5) sinais de controle (“A”, “B”, “C”, “AB” e “ABC”). Analisando estas duas figuras, é fácil perceber como atua o circuito lógico de correção de erro de sincronismo de bit.

No circuito de correção do  $MSB/\overline{MSB}$ , quando o sinal de controle “A” estiver em nível alto (região 1), o sinal de saída “MSB” é substituído pelo sinal inverso do 4SB (seria o  $\overline{4SB}$ ). Isto irá ocorrer apenas no período de transição do MSB. Durante o restante do tempo, o sinal de saída será o próprio sinal de entrada MSB.

No circuito de correção do  $2SB/\overline{2SB}$ , quando o sinal de controle “A” estiver em nível alto (região 1), o sinal de saída “2SB” é substituído pelo sinal do 4SB. Quando o sinal de controle “B” estiver em nível alto (regiões 4 e 5), o sinal de saída “2SB” é substituído pelo sinal inverso do 4SB (seria o  $\overline{4SB}$ ). Quando os sinais de controle “A” ou “B” estiverem em nível alto, o sinal de controle “AB” também estará em nível alto (regiões 1, 4 e 5) e, conseqüentemente, o sinal de entrada 2SB estará desabilitado. Isto irá ocorrer apenas no período de transição do 2SB. Durante o restante do tempo, o sinal de saída será o próprio 2SB.

No circuito de correção do  $3SB/\overline{3SB}$ , quando o sinal de controle “AB” estiver em nível alto (regiões 1, 4 e 5), o sinal de saída “3SB” é substituído pelo sinal do 4SB. Quando o sinal de controle “C” estiver em nível alto (regiões 2, 3, 6 e 7), o sinal de saída “3SB” é substituído pelo sinal inverso do 4SB (seria o  $\overline{4SB}$ ). Quando os sinais de controle “AB” ou “C” estiverem em nível alto, o sinal de controle “ABC” também estará em nível alto (regiões 1, 2, 3, 4, 5, 6 e 7) e, conseqüentemente, o sinal de entrada 3SB estará desabilitado. Isto irá ocorrer apenas no período de transição do 3SB. Durante o restante do tempo, o sinal de saída será o próprio 3SB.

O Circuito utilizado nas portas “AND” e na porta “OR” de duas (2) entradas (operação feita por uma porta “NAND”) é o da Fig. 3.28 e o circuito utilizado na porta “OR” de três (3)

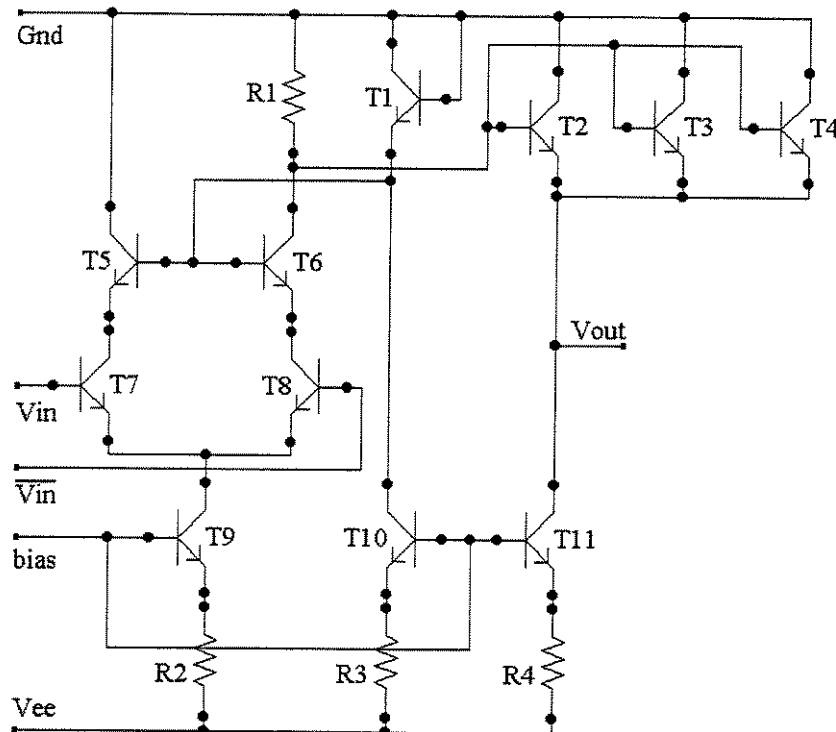


Figura 3.37 – Porta de Saída do Conversor A/D de Bits

entradas (operação também feita por uma porta ‘NAND’) é o da Fig. 3.29.

### 3.10 – Porta de Saída:

O  $MSB/\overline{MSB}$ ,  $2SB/\overline{2SB}$  e  $3SB/\overline{3SB}$  (após receberem a correção de sincronismo de bit) e o  $4SB/\overline{4SB}$ , vão para as portas de saída. Os bits menos significativos ( $5SB/\overline{5SB}$ ,  $6SB/\overline{6SB}$ ,  $7SB/\overline{7SB}$  e  $LSB/\overline{LSB}$ ), da saída da estrutura decodificadora tipo “ROM”, também vão para as portas de saída. O circuito da porta de saída (veja a Fig. 3.37) tem por objetivo servir de “buffer” e ajustar o sinal de saída aos níveis “ECL” em código binário. Os sinais de saída do conversor A/D não são diferenciais, contudo, todos os sinais internos do conversor A/D, até a entrada da porta de saída, são diferenciais.

A Fig. 3.38 mostra os sinais de saída resultantes da simulação deste conversor A/D de 8 bits tipo “folding” com interpolação e correção de erro.

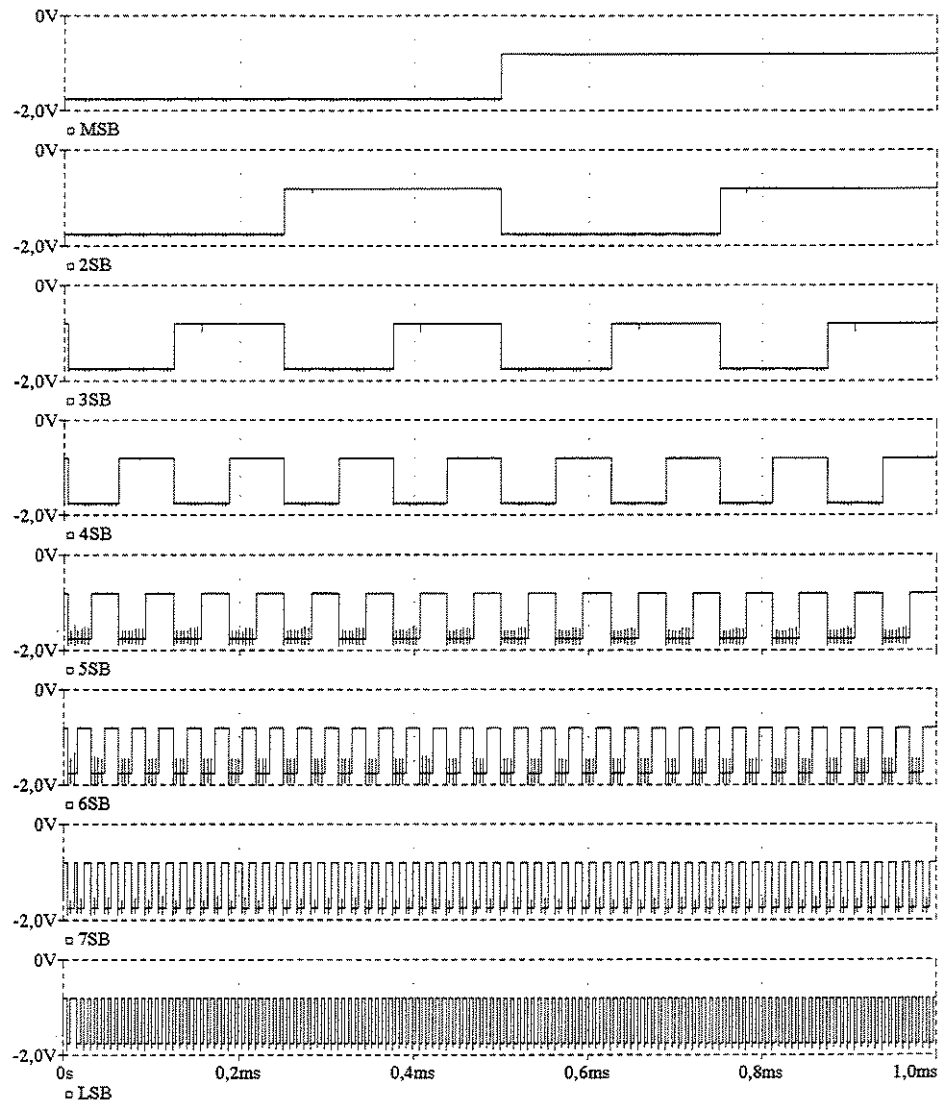


Figura 3.38 – Sinais de Saída

### 3.11 – Resultados Obtidos em Simulação:

#### 3.11.1 – Características Gerais:

A seguir serão apresentados os resultados do conversor A/D, obtidos em simulação. Para tanto, foi colocado na saída do conversor A/D, um conversor D/A, tipo R-2R (veja a Fig. 3.39), para reconstrução do sinal analógico de entrada. Na avaliação do conversor A/D foram considerados os parâmetros mais utilizados na referência bibliográfica, que são: o DNL, o INL, o número efetivo de bits [01 – 22] e o teste da frequência de batimento [26].

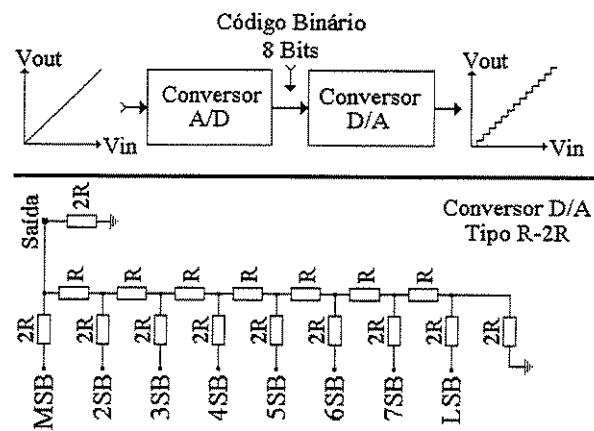


Figura 3.39 – Conversor A/D e D/A Interligados

3.11.2 – Consumo de Potência:

Tabela 3.8 – Consumo de Potência

| Descrição/Temperatura | 0°C    | 27°C   | 70°C   |
|-----------------------|--------|--------|--------|
| Sem Carga             | 585 mW | 612 mW | 657 mW |
| Com Carga *           | 747 mW | 776 mW | 860 mW |

(\*) Carga de 50 Ω em Relação a – 2 V

3.11.3 – Erro de Quantização:

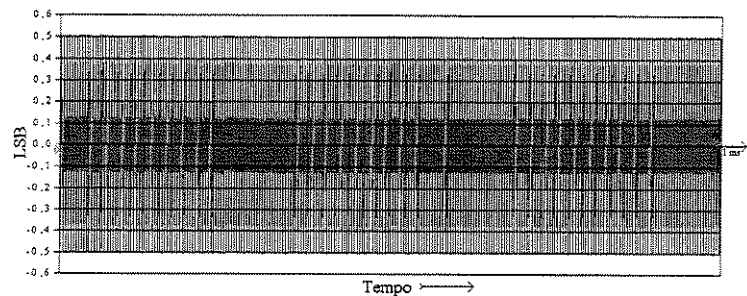


Figura 3.40 – Erro de Quantização

3.11.4 – Resultados Estáticos:

Tabela 3.9 – Resultados Estáticos

| Sinais ⇒ F <sub>Vin</sub> = 1 kHz e F <sub>Clock</sub> = 256 kHz |           |            |           |
|------------------------------------------------------------------|-----------|------------|-----------|
| Medidas/Temperatura                                              | 0°C       | 27°C       | 70°C      |
| Não Linearidade Diferencial - DNL                                | <±0,1 LSB | <±0,01 LSB | <±0,3 LSB |
| Não Linearidade Integral - INL                                   | <±0,2 LSB | <±0,02 LSB | <±0,5 LSB |

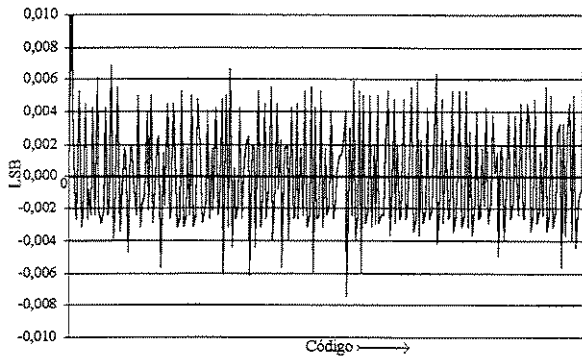


Figura 3.41 (a) – DNL

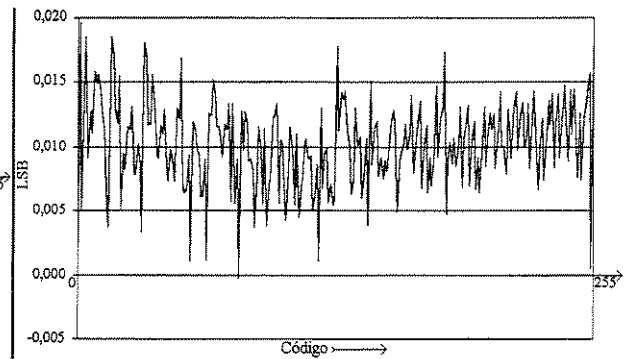


Figura 3.41 (b) – INL

### 3.11.5 – Resultados Dinâmicos:

#### 3.11.5.1 – Número Efetivo de Bits:

Tabela 3.10 – Resultados Dinâmicos

|                                       |                                             |
|---------------------------------------|---------------------------------------------|
| Máxima Frequência do Sinal de Entrada | 20 MHz (para Número Efetivo de Bits = 7,5)  |
| Máxima Frequência do Sinal de “Clock” | 400 MHz (para Número Efetivo de Bits = 7,5) |

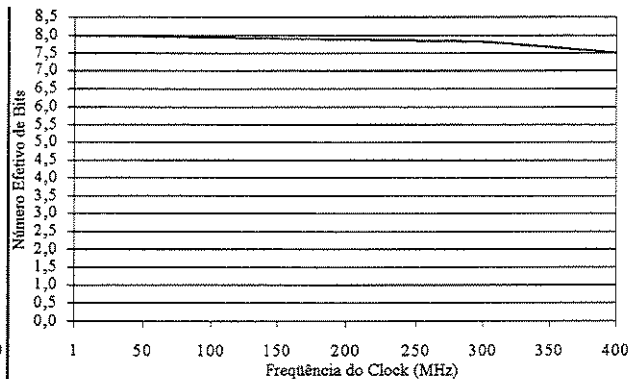
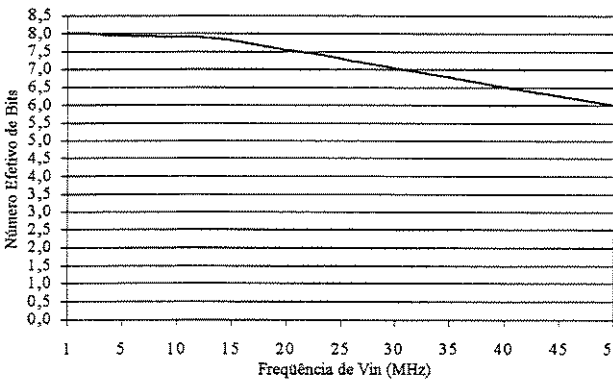


Figura 3.42 – Número Efetivo de Bits

#### 3.11.5.2 – Teste da Frequência de Batimento:

Sinais  $\Rightarrow f_{in} = 5.078.125 \text{ Hz}$  e  $f_s = 5.000.000 \text{ Hz}$  e  $f_{out} = d_f = 78.125 \text{ Hz}$

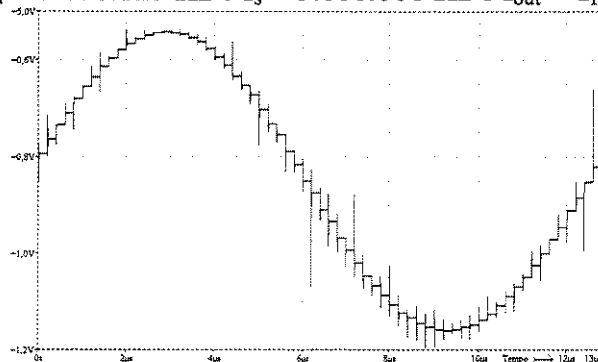


Figura 3.43 – Teste da Frequência de Batimento

### 3.11.6 – “Monte Carlo”:

O programa utilizado para simulação deste conversor A/D apresenta um recurso chamado simulação “Monte Carlo”, que permite fazer uma análise estatística no circuito simulado. Isto é, pode-se dar tolerâncias (%) aos parâmetros dos modelos utilizados e o programa simulador faz múltiplas simulações utilizando-se destas tolerâncias. Um fator importante é que estas tolerâncias são aplicadas independentemente dos componentes que utilizam o mesmo modelo. Por exemplo: se no circuito existem dois resistores iguais a 1 k $\Omega$  e com o mesmo modelo, o programa aplica a tolerância determinada para o parâmetro especificado (no caso o valor do resistor) de forma que, durante a simulação, um resistor possa ter o valor de 950  $\Omega$  e o outro de 1050  $\Omega$ .

Como o recurso requer múltiplas simulações, é aplicada uma distribuição estatística dos valores para cada simulação. Este recurso permite avaliar de forma mais real o comportamento do circuito porque, na prática, os componentes apresentam pequenas variações em função da própria construção do circuito integrado.

Na aplicação deste recurso para análise deste conversor A/D (que requer longo tempo de simulação), foram consideradas principalmente as variações que poderiam ocorrer entre os  $V_{BE}$  dos transistores. Para tanto, utilizou-se a variação do parâmetro “IS” do modelo de transistor, de forma que refletisse um provável descasamento entre os  $V_{BE}$  dos transistores. Para transistores com as mesmas correntes, teríamos:  $\Delta V_{BE} = V_T \ln(IS_2/IS_1)$ . Dentro da forma geral do recurso, foram realizadas 3 simulações: a primeira utilizou os valores nominais dos parâmetros dos modelos e as subseqüentes fizeram a variação estatística destes parâmetros. A Tabela 3.11 mostra os resultados comparativos.

Tabela 3.11 – Resultados da Simulação “Monte Carlo”

| Medidas: 27°C / $\Delta V_{BE} = 1$ mV | Simulação Nominal | 1ª Simulação com Variação de IS | 2ª Simulação com Variação de IS |
|----------------------------------------|-------------------|---------------------------------|---------------------------------|
| DNL                                    | $<\pm 0,01$ LSB   | $<\pm 0,50$ LSB                 | $<\pm 0,50$ LSB                 |
| INL                                    | $<\pm 0,02$ LSB   | $<\pm 0,50$ LSB                 | $<\pm 0,50$ LSB                 |

Tabela 3.12 – Correção de Erro de Código

| Tempo     | Sinais      |               |         |               |             | Sem Correção |     |     | Com Correção |     |     | Comentários                   |
|-----------|-------------|---------------|---------|---------------|-------------|--------------|-----|-----|--------------|-----|-----|-------------------------------|
|           | 67,5°/67,5° | 56,25°/56,25° | 45°/45° | 33,75°/33,75° | 22,5°/22,5° | 3'           | 4'  | 5'  | 3            | 4   | 5   |                               |
| 8 μs      | 0           | 0             | 0       | 0             | ↑ 1         | 1            | 0   | 0   | 1            | 0   | 0   |                               |
| 9,772 μs  | 0           | 0             | 0       | ↑ 1           | 1           | ↓ 0          | ↑ 1 | 0   | ↓ 0          | ↑ 1 | 0   |                               |
| 13,684 μs | 0           | 0             | 0       | 1             | 1           | 0            | 1   | 0   | 0            | 1   | 0   | Ainda não ocorreu nenhum erro |
| 17,597 μs | 0           | ↑ 1           | 0       | 1             | 1           | 0            | 1   | ↑ 1 | 0            | ↓ 0 | ↑ 1 | Ocorreu o Erro 1              |
| 21,491 μs | ↑ 1         | 1             | 1       | 1             | 1           | 0            | ↓ 0 | ↓ 0 | 0            | 0   | ↓ 0 |                               |

Obs.: Os sinais “22,5°/22,5°”, “33,75°/33,75°”, “45°/45°”, “56,25°/56,25°” e “67,5°/67,5°” são oriundos das portas “Ou-Exclusivo” com “Latch”, veja a Tabela 3.4 e a Fig. 3.21. Os sinais 3, 4 e 5 são oriundos das saídas das portas “Ou-Exclusivo” com correção de erro e devem endereçar apenas uma linha da “ROM”, veja a Tabela 3.5 e a Fig. 3.23. Os sinais 3', 4' e 5' seriam oriundos das saídas das portas “Ou-Exclusivo” sem a implementação da correção de erro.

### 3.11.7 – Correção de Erro de Código:

No teste de correção de erro, foi forçada a ocorrência do “erro 1” (veja a Tabela 3.6). Observando-se a Tabela 3.12 e a Fig. 3.44, é possível acompanhar a correção do erro. No instante de 13,684 μs, o sinal 45°/45° deveria ter ido ao nível “1”, considerando que o nível do sinal de entrada estivesse aumentando, o que não é necessariamente obrigatório e, conseqüentemente, o erro ainda não poderia ser caracterizado. No instante de 17,597 μs, o sinal 56,25°/56,25° foi ao nível “1” e o sinal 45°/45° continuou em nível “0”, o que caracterizou o erro 1. Neste momento, se não houvesse a correção de erro, ocorreria o endereçamento de duas (2) linhas da estrutura tipo “ROM”. A expressão de 4' seria igual a  $(45^\circ/45^\circ \oplus 33,75^\circ/33,75^\circ) \rightarrow “1”$  e a expressão de 5' seria igual a:  $(56,25^\circ/56,25^\circ \oplus 45^\circ/45^\circ) \rightarrow “1”$ . Aplicando-se a correção, a expressão de 4 fica igual a  $(45^\circ/45^\circ \oplus 33,75^\circ/33,75^\circ) \text{ AND } (56,25^\circ/56,25^\circ \oplus 22,5^\circ/22,5^\circ) \rightarrow “1” \text{ AND } “0” \rightarrow “0”$ . De forma semelhante, a expressão de 5 ficaria igual a  $(56,25^\circ/56,25^\circ \oplus 45^\circ/45^\circ) \text{ AND } (67,5^\circ/67,5^\circ \oplus 33,75^\circ/33,75^\circ) \rightarrow “1” \text{ AND } “1” \rightarrow “1”$ . Diante destes resultados, verifica-se que

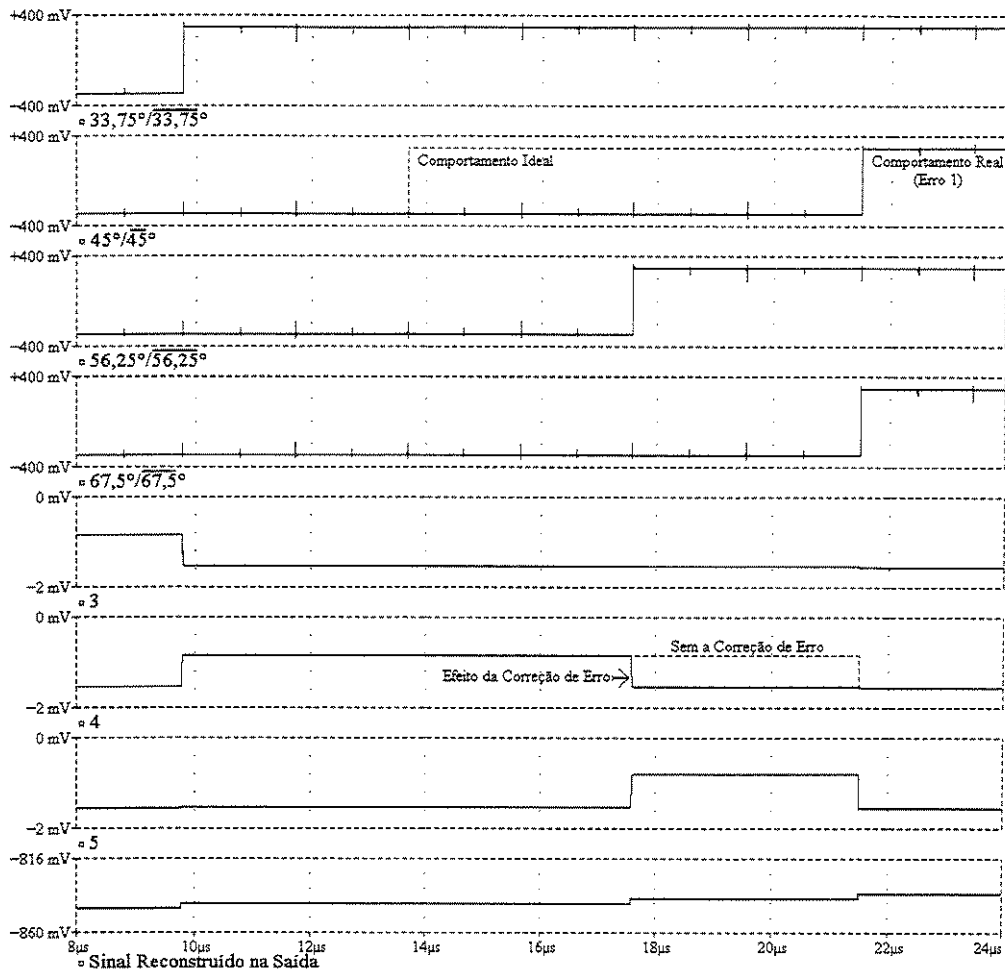


Figura 3.44 – Correção de Erro

a atuação do circuito de correção de erro corrige o sinal 4, o que impede o endereçamento de duas linhas da estrutura tipo “ROM”.

### 3.11.8 – Análise dos Resultados:

Os resultados obtidos por simulação devem ser entendidos apenas como uma previsão de comportamento e desempenho do conversor A/D. Considerando-se que estes resultados não foram obtidos do circuito integrado real, os mesmos não contêm a influência das imperfeições existentes e inerentes ao circuito real e, desta forma, apresentam alguns valores otimizados.

Os resultados obtidos através da simulação são influenciados por dois fatores: a razão entre a frequência do sinal de “clock” e a frequência do sinal de entrada ( $F_{\text{Clock}}/F_{\text{Vin}}$ ) e o valor do

passo (“step”) utilizado na simulação. A razão entre a frequência do sinal de “clock” e a frequência do sinal de entrada ( $F_{\text{Clock}}/F_{\text{Vin}}$ ) determina a precisão do resultado obtido e o respectivo tempo de simulação. Por exemplo, se a razão  $F_{\text{Clock}}/F_{\text{Vin}}$  for igual a 256 ( $2^8$ ), a precisão das medidas obtidas será de 1 LSB porque, neste caso, haverá um sinal de clock para cada 1 LSB. Isto pode ser facilmente demonstrado, supondo a eventual ocorrência de um meta-estado em algum “latch”, o que causaria a perda de 1 LSB no código digital de saída. Se a razão  $F_{\text{Clock}}/F_{\text{Vin}}$  for igual a 2560 ( $10 \times 2^8$ ), a precisão das medidas será igualmente de 0,1 LSB porque a eventual ocorrência de um meta-estado em algum “latch” provocaria a perda de apenas 0,1 LSB no código digital de saída. A razão  $F_{\text{Clock}}/F_{\text{Vin}}$  também influencia diretamente o tempo de simulação. Supondo um sinal de entrada de 1 KHz e uma  $F_{\text{Clock}}$  (igual a  $1/T_{\text{Clock}}$ ) de 256 KHz ( $F_{\text{Clock}}/F_{\text{Vin}} = 256$ ), o tempo de simulação do conversor A/D (por exemplo, para a simulação da “análise de transiente” para 1 ms) fica em aproximadamente 15:30 horas (em um computador Pentium II – 266 MHz). Dobrando a  $F_{\text{Clock}}$  ( $F_{\text{Clock}}/F_{\text{Vin}} = 512$ ), o número de pontos a ser calculado aumenta e, conseqüentemente, o tempo de simulação fica em cerca de 25:35 horas. De forma semelhante, diminuindo o passo (ou “step”), o número de pontos a ser calculado aumenta e, conseqüentemente, o tempo de simulação também aumenta. Dentro desta considerações, a maioria dos resultados de simulação mostrados foram realizados prevendo-se uma precisão de  $\pm 0,5$  LSB.

O conversor A/D proposto utiliza a arquitetura “folding” com interpolação, opera com uma fonte simples de alimentação de – 5 V, uma fonte de referência de – 1,125 V e consome cerca de 612 mW (sem carga a 27°C). Os valores de linearidade (DNL e INL), obtidos em simulação, podem ser considerados bons (o ideal é ficar abaixo de 1 LSB).

A simulação Monte Carlo (que pressupõe a variação dos  $V_{\text{BE}}$ ), através da análise estatística, permite avaliar de forma mais real o comportamento do conversor A/D. Os resultados,

que podem ser considerados os mais importantes nesta análise do conversor A/D, demonstraram que houve aumento no valor do DNL e INL com a variação estatística dos  $V_{BE}$ . Na precisão da interpolação, houve uma variação de 0,25 LSB ( $\Delta V_{BE} = 1 \text{ mV}$ ). É importante verificar que, conforme [28], a técnica “folding” com interpolação diminui a necessidade dos ajustes entre os  $V_{BE}$ . De qualquer forma, a provável variação entre os transistores na fabricação do circuito integrado não prejudicará de forma significativa o desempenho do conversor A/D.

Conforme a definição de largura de banda com resolução de N-bit [05] e o valor obtido do número efetivo de bits, conclui-se que a largura de banda prevista para este conversor é de 20 MHz e pode trabalhar com uma frequência do sinal de “clock” de aproximadamente 400 MHz. De fato, o conceito da técnica “folding” com interpolação é de um conversor A/D de baixo consumo, com um circuito menor em relação a técnica “flash”, alta taxa de conversão, porém tem o inconveniente de ter uma performance dinâmica limitada em função da razão de “folding” do amplificador “folding” [29]. A solução que pode ser aplicada é aumentar o número de amplificadores “folding” em paralelo e diminuir a razão de “folding” em cada um deles. Esta solução tende a aumentar a complexidade do conversor A/D, aproximando-a da complexidade da técnica “flash”.

De acordo com o esperado, a correção de erro do código circular interno do conversor A/D, corrige a decisão errada de algum “latch”; porém, não corrige um eventual meta-estado do “latch”. Neste caso, o nível lógico inválido será propagado e resultará em um correspondente código errado na saída, causando uma descontinuidade, ou “glitch”, na forma de onda do sinal reconstruído na saída.

A Fig. 3.38 (veja o item 3.10) mostra um exemplo dos sinais de saída obtidos através de uma simulação de “análise de transiente” e a Tabela 3.13 mostra o resumo das características especificadas e simuladas do conversor A/D.

### 3.11.9 – Especificações Gerais do Conversor A/D (Resultados Simulados):

Tabela 3.13 – Especificações do Conversor A/D (Simulado)

|                              |                                                               |
|------------------------------|---------------------------------------------------------------|
| Resolução                    | 8 bits                                                        |
| Número Efetivo de Bits       | 7,9 bits                                                      |
| Largura de Banda de $V_{in}$ | 20 MHz (Número Efetivo de Bits = 7,5 e $F_{Clock} = 100$ MHz) |
| Frequência de Amostragem     | 400 MHz (Número Efetivo de Bits = 7,5 e $F_{Vin} = 10$ KHz)   |
| Tempo de Conversão           | 1 $T_{clock}$                                                 |
| Voltagem de Alimentação      | -5 V                                                          |
| Voltagem de $V_{in}$         | -1,0625 a -0,0625                                             |
| DNL                          | <0,01 LSB                                                     |
| INL                          | <0,02 LSB                                                     |
| Código de Saída Digital      | Binário - ECL 10 000                                          |
| Número de Elementos          | 2675                                                          |
| Transistores                 | 2087                                                          |
| Resistores                   | 588                                                           |

### 3.12 – “Layout”:

A tecnologia utilizada foi de 0,8  $\mu\text{m}$  - BiCMOS - Substrato p (“Byb”) da “Austria Mikro Systeme International AG ”– AMS. A adoção desta tecnologia implica na utilização dos parâmetros de processo (para simulação) e nas regras de projeto (para elaboração do “layout”).

O “layout” foi elaborado no programa “IC Station”, v8.6\_3.5, da “Mentor Graphics” e também foi utilizado o “AMS HIT - Kit Version 2.50”. O circuito ficou com uma área de 7,67  $\text{mm}^2$  (4086,95  $\mu\text{m}$  x 1876,05  $\mu\text{m}$ ); esta área inclui os “pads”.

A Tabela 3.14 mostra a relação básica das “layers” do processo e a Tabela 3.15 mostra os parâmetros de processo para simulação do transistor bipolar.

A Fig. 3.45 mostra a configuração básica do transistor bipolar e a Fig. 3.46 mostra a configuração básica de um resistor, conforme as regras de projeto do processo.

A Fig. 3.47 mostra o “layout” final do circuito do conversor A/D de 8 bits e nos anexos foi colocado o “layout” de alguns dos circuitos específicos utilizados.

Tabela 3.14 – Relação Básica das “Layers” do Processo

| Nº | Layer   | Design Kit | Tipo      | Descrição                                  |
|----|---------|------------|-----------|--------------------------------------------|
|    | AMS     | Mentor     |           |                                            |
| 1  | NTUB    | NTM        | NTUB      | poço n                                     |
| 2  | PBURIED | PBL        | BIL       | camada enterrada p+                        |
| 3  | NBURIED | BLM        | BLM       | camada Enterrada n+                        |
| 4  | DIFF    | FMK        | DIFFUSION | difusão                                    |
| 5  |         |            |           |                                            |
| 6  | FIMP    | NFM        | F-IMPLANT | n-field implant layer                      |
| 7  |         |            |           |                                            |
| 8  | SK_DEF  | CLM        | CLM       | difusão n+ profunda (sinker - coletor)     |
| 9  | B_DEF   | BIL        | PBL       | evita o NPLUS na base/emissor              |
| 10 | POLY1   | GMK        | POLY      | Poly 1 - 2 x 2 $\mu\text{m}$ - de resistor |
| 11 | E_CUT   | EMK        | EMK       | abertura de emissor                        |
| 12 | NPLUS   | ASM        | ARSENIC   | n+ implant layer - arsênico                |
| 13 | PPLUS   | BSM        | BORON     | p+ implant layer - boro                    |
| 14 | POLY2   | PM2        | POLY2     | Poly 2 - 2 x 2 $\mu\text{m}$ - de resistor |
| 15 | HRES    |            | HRM       | não disponível                             |
| 16 | CONT    | CMK        | CONTACT   | contato                                    |
| 17 | MET1    | MM1        | METAL1    | metal 1 - 2,4 $\mu\text{m}$                |
| 18 | VIA     | VMK        | VIA       | via                                        |
| 19 | MET2    | MM2        | METAL2    | metal 2 - 2,4 $\mu\text{m}$                |
| 20 | PAD     | NMK        | PADWINDOW | padwindow                                  |

Tabela 3.15 – Parâmetros de Processo do Transistor Bipolar

| TYPICAL MEAN CONDITION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>* Vertical poly-emitter single base contact NPN transistor: [ C-E-B ].</p> <p>* Emitter width=0.8<math>\mu\text{m}</math>, parameters are referred to emitter length L0=1<math>\mu\text{m}</math>.</p> <p>* The emitter length LE must be specified as the device area AREA=LE/L0.</p> <p>.SUBCKT NPN111 C B E S PARAM: AREA=0</p> <p>Q1 C B E S NPN111 (AREA)</p> <p>.MODEL NPN111 NPN</p> <p>+IS={1.550e-18+3.000e-19/AREA} NF=1.000e+00</p> <p>+IKF={2.070e-03+7.000e-04/AREA} BF={1.060e+02+1.850e+01/AREA}</p> <p>+ISE={1.100e-16+4.180e-15/AREA} NE=3.500e+00 VAF=3.230e+01</p> <p>+IKR=1.300e-03 BR=1.000e+01 NR=1.010e+00</p> <p>+ISC=1.600e-18 NC=1.300e+00 VAR=3.670e+00</p> <p>+RBM=9.000e+02 TRM1=3.200e-03 RB=2.660e+03 TRB1=3.200e-03</p> <p>+RE={2.630e+01+1.120e+00*AREA} AF=1.600e+00</p> <p>+RC={1.0/(1.0/2.150e+03+1.0/(2.010e+02*AREA)))} KF=1.000e-12</p> <p>+TF=8.240e-12 XTF=8.000e+00 ITF=2.450e-03</p> <p>+VTF=1.760e+00 PTF=0.000e+00 TR=1.000e-09 XCJC=0.000e+00</p> <p>+EG=1.160e+00 XTI=3.300e+00 XTB=1.800e+00 FC=7.500e-01</p> <p>+CJE={3.480e-15+1.570e-15/AREA} VJE=9.000e-01 MJE=5.000e-01</p> <p>+CJC={1.370e-15+6.650e-15/AREA} VJC=8.000e-01 MJC=3.700e-01</p> <p>+CJS={4.580e-15+3.280e-14/AREA} VJS=8.000e-01 MJS=3.700e-01</p> <p>.ENDS NPN111</p> |
| WORST CASE TOLERANCES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <p>* Basic Gummel-Poon parameter: AREA dependence:</p> <p>* 1.070e-18 &lt; IS &lt; 2.183e-18 2.070e-19 &lt; ISL &lt; 4.225e-19</p> <p>* 7.590e-17 &lt; ISE &lt; 1.549e-16 2.884e-15 &lt; ISEL &lt; 5.887e-15</p> <p>* 7.729e+01 &lt; BF &lt; 1.767e+02 1.349e+01 &lt; BFL &lt; 3.083e+01</p> <p>* 1.900e+01 &lt; VAF &lt; 4.180e+01</p> <p>* 1.194e-03 &lt; IKF &lt; 3.185e-03 4.038e-04 &lt; IKFL &lt; 1.077e-03</p> <p>* 5.921e+02 &lt; RBM &lt; 1.066e+03</p> <p>* 1.750e+03 &lt; RB &lt; 3.150e+03</p> <p>* 2.032e+01 &lt; RE &lt; 4.184e+01 8.655e-01 &lt; REL &lt; 1.782e+00</p> <p>* 1.891e+03 &lt; RC &lt; 2.591e+03 1.768e+02 &lt; RCL &lt; 2.422e+02</p> <p>* 7.289e-12 &lt; TF &lt; 9.476e-12</p> <p>* 3.000e-15 &lt; CJE &lt; 3.900e-15 1.353e-15 &lt; CJEL &lt; 1.759e-15</p> <p>* 1.310e-15 &lt; CJC &lt; 1.489e-15 6.361e-15 &lt; CJCL &lt; 7.228e-15</p> <p>* 3.857e-15 &lt; CJS &lt; 5.303e-15 2.762e-14 &lt; CJSL &lt; 3.798e-14</p>                                                                                                                                                                                                                                                                                                                                        |

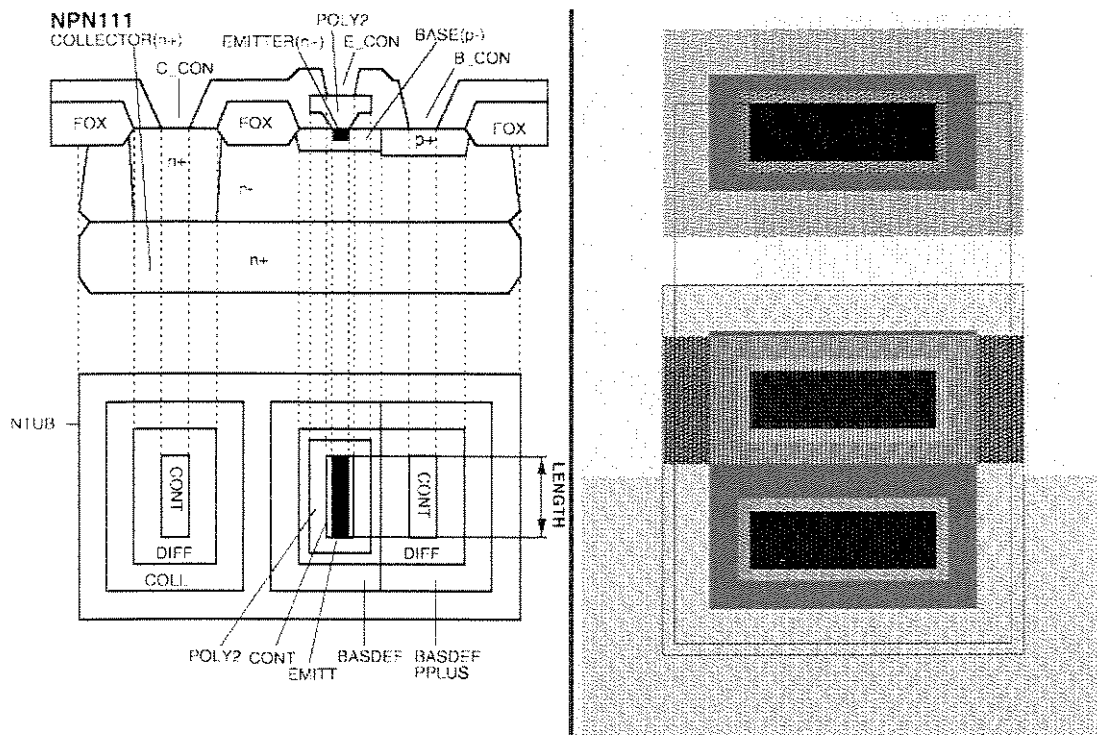


Figura 3.45 – Configuração Básica do Transistor Bipolar

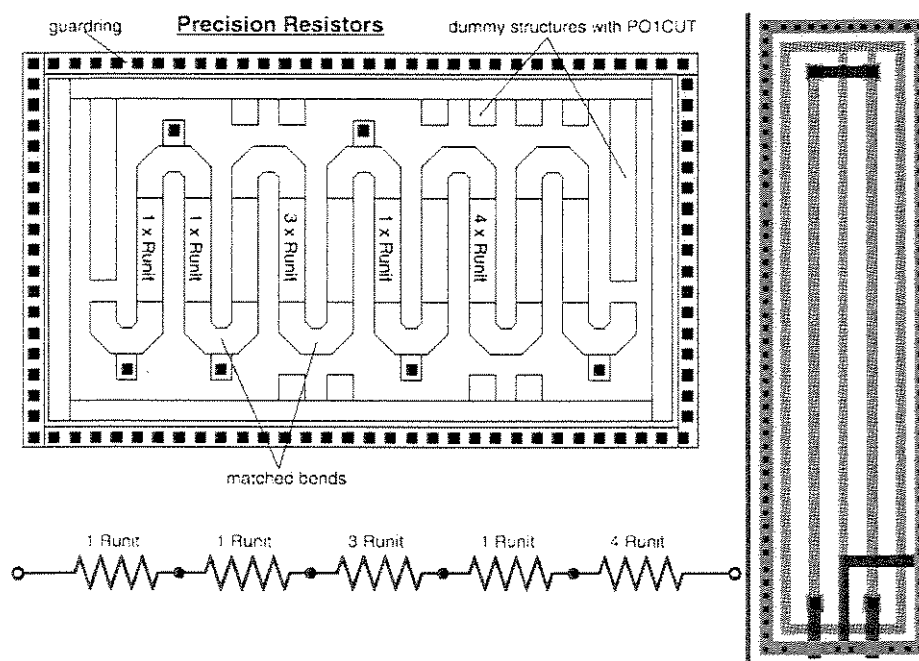


Figura 3.46 – Configuração Básica de Um Resistor

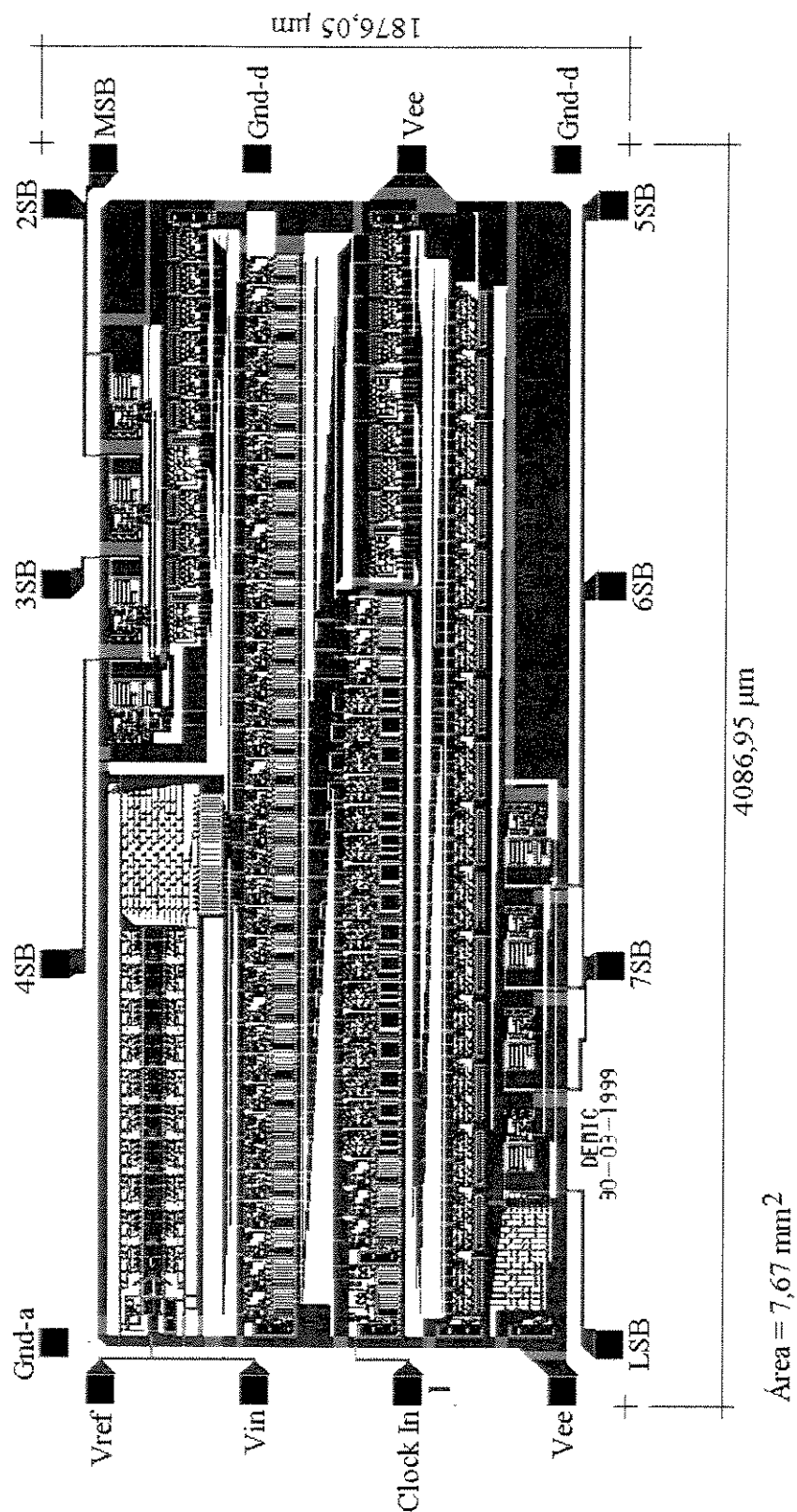


Figura 3.47 – “Layout” Final do Conversor A/D de 8 Bits

## Capítulo 4

### Resultados

#### *4.1 – Avaliação dos Conversores A/D*

##### *4.1.1 – Conversores A/D:*

Foram construídos três conversores A/D para avaliação das novas técnicas de interpolação e correção de erro. O primeiro foi um conversor A/D de 6 bits para avaliar a proposta inicial de interpolação. O segundo foi um conversor A/D de 8 bits que utiliza a topologia convencional para avaliar a proposta final de interpolação. O terceiro foi um conversor A/D de 8 bits que faz uso de uma nova topologia, com a proposta final de interpolação, a correção de erro de código e o sincronismo de bits.

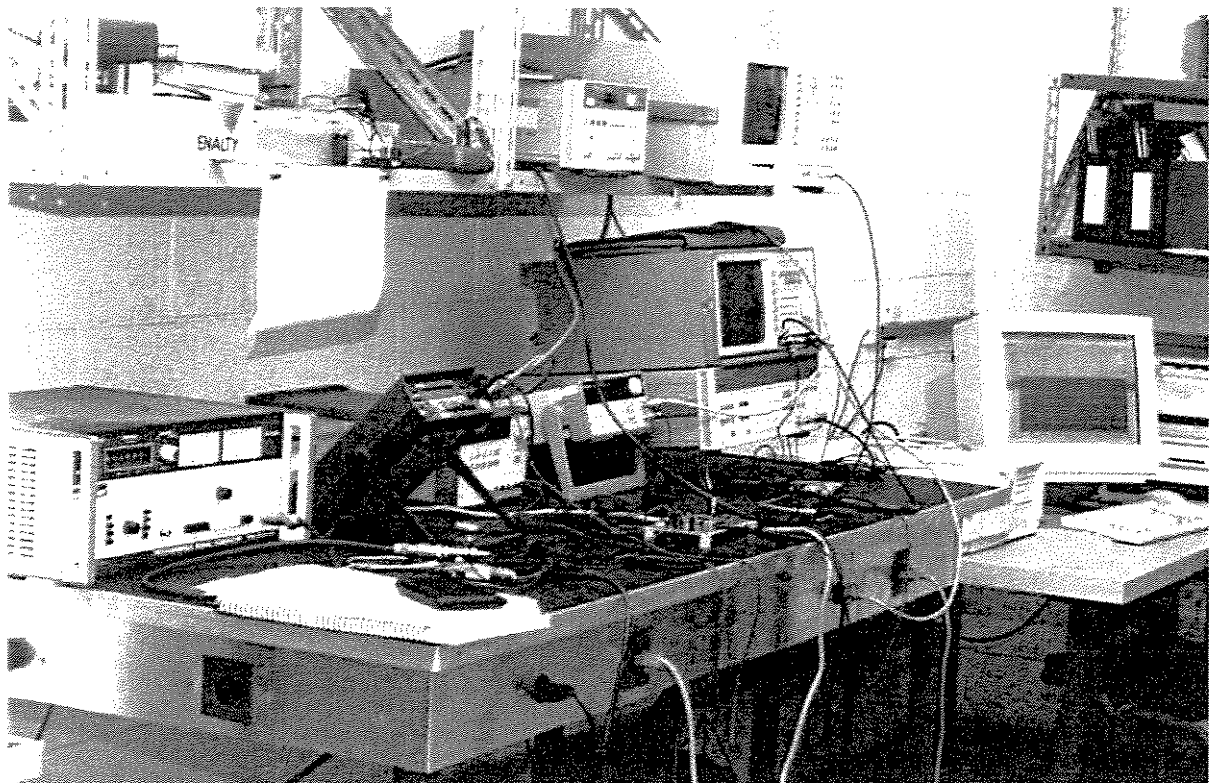


Figura 4.1 – *Laboratório de Instrumentação e Sensores - LIS*

### 4.1.2 – Laboratório:

As avaliações dos três conversores A/D construídos foram realizadas no “Laboratório de Instrumentação e Sensores - LIS” do Departamento de Eletrônica e Microeletrônica (DEMIC) que pode ser visto na Fig. 4.1.

## 4.2 – Conversor A/D de 6 bits:

### 4.2.1 – Descrição:

A seguir, serão apresentados os resultados medidos do conversor A/D de 6 bits que contém a proposta inicial de interpolação, não possui circuito de correção de erro e nem sincronismo de bits. O circuito integrado (CI) foi implementado em tecnologia BiCMOS de  $1,2\ \mu\text{m}$ ,  $F_T$  de 8 GHz, p-substrato, duplo metal, duplo “polysilicon”, processo da AMS (Austria Mikro Systeme International AG). Este circuito integrado (FAPESP\_50) foi construído na rodada “A97-18” de 21 de novembro de 1997 da AMS, através do PMU / FAPESP e foi recebido em julho de 1998.

### 4.2.2 – Micrografia do Circuito Integrado:

A Fig. 4.2 mostra a micrografia do circuito integrado do conversor A/D de 6 bits para um

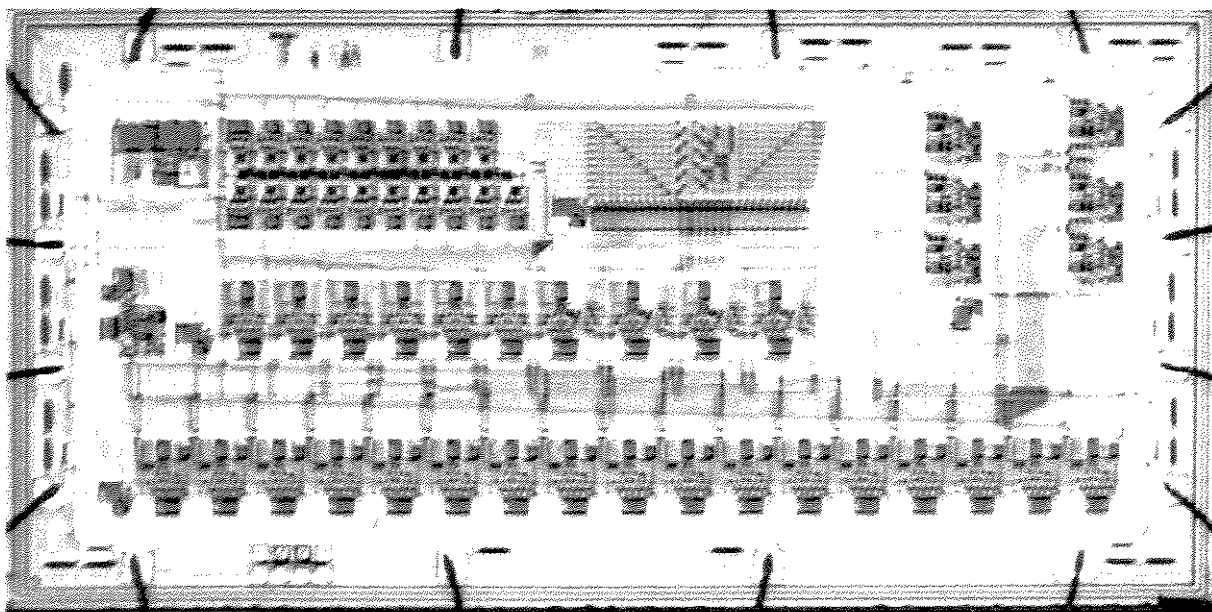


Figura 4.2 – Micrografia do Conversor A/D de 6 bits

encapsulamento de 16 pinos (DIL). A área total do “chip” é de  $1835,80 \mu\text{m} \times 3798,60 \mu\text{m} = 6,97 \text{ mm}^2$ . A área ficou maior do que o previsto em função da limitada experiência na elaboração de “layout”.

#### 4.2.3 – Sistema de Teste:

A Fig. 4.3 mostra o sistema de teste aplicado e a Tabela 4.1 a relação dos equipamentos e software utilizados. O conversor D/A utilizado foi o DAC 08CN (Philips Semiconductors). As medidas realizadas tiveram por objetivo avaliar a performance estática (DNL e INL) e dinâmica (número efetivo de bits) do conversor A/D de 6 bits.

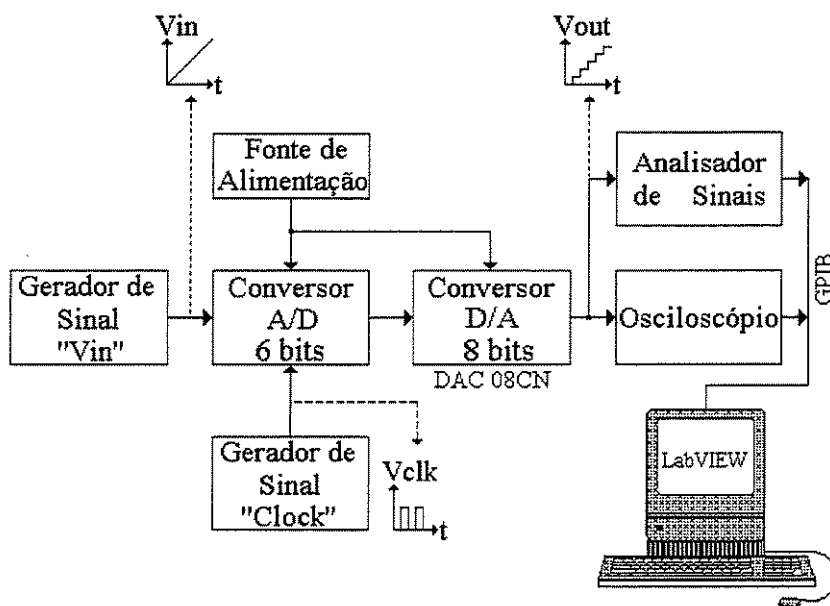


Figura 4.3 – Sistema de Teste

Tabela 4.1 – Relação dos Equipamentos e Software Utilizados

| Descrição            | Marca                | Modelo           |
|----------------------|----------------------|------------------|
| Osciloscópio         | Hewlett Packard      | 54503A – 500 MHz |
| Analizador de Sinais | Hewlett Packard      | 35660A – 100 KHz |
| Gerador de Sinais    | Hewlett Packard      | 33120A – 15 MHz  |
| Gerador de Sinais    | Hewlett Packard      | 3314A – 20 MHz   |
| Fonte de Alimentação | Hewlett Packard      | E3631A           |
| Aquisição de Dados   | National Instruments | LabVIEW 5.0      |

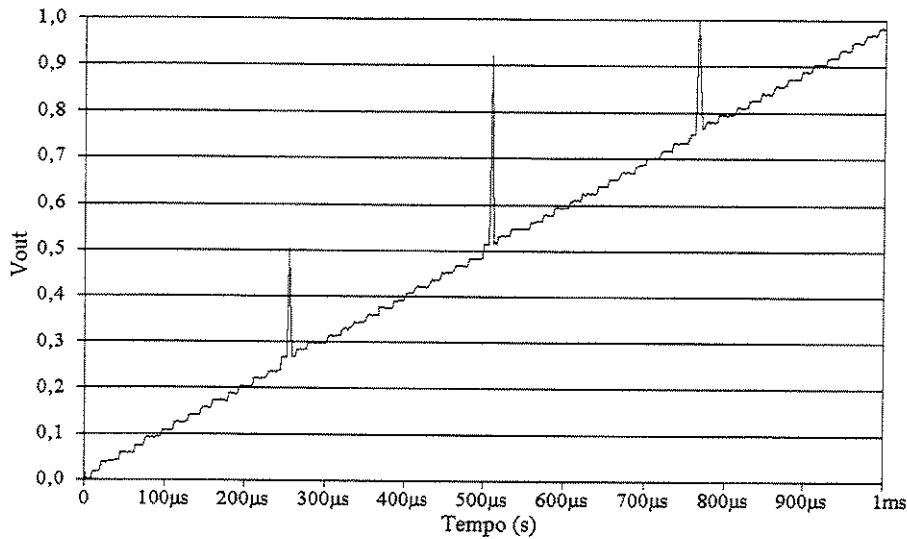


Figura 4.4 – Sinal de Saída do Conversor A/D de 6 Bits

#### 4.2.4 – Sinal de Saída Reconstruído:

A Fig. 4.4 mostra um exemplo do sinal de saída do conversor A/D de 6 bits reconstruído pelo conversor D/A, para um sinal de entrada de 1 KHz (tipo rampa) e a frequência do sinal de “clock” de 10 MHz. Os “glitches” (transientes associados com a troca de código) presentes no sinal são precisamente resultados da falta do circuito de sincronismo de bits entre o 3SB, o 2SB e o MSB.

#### 4.2.5 – Especificações Gerais do Conversor A/D de 6 Bits:

A Tabela 4.2 mostra um resumo das características e dos resultados medidos na avaliação do conversor A/D de 6 bits. As medidas foram feitas utilizando o sistema de teste mostrado na Fig. 4.3 e foram realizadas desprezando os efeitos dos “glitches”. Os valores do DNL e INL ficaram acima dos obtidos em simulação (0,03 LSB e 0,11 LSB respectivamente). Isto ocorreu em função das relações de áreas adotadas para os transistores de interpolação (veja a análise no item 4.5.1). A máxima frequência para o sinal de entrada e a máxima frequência para o sinal de “clock” ficaram abaixo das obtidas em simulação (10 MHz e 320 MHz respectivamente) em função dos parasitas do “layout” e da limitada experiência na elaboração de “layout” para

Tabela 4.2 – Especificações Gerais do Conversor A/D de 6 Bits

|                               |                                                                    |
|-------------------------------|--------------------------------------------------------------------|
| Resolução                     | 6 bits                                                             |
| Sinal de Entrada              | -1,0625 V a -0,0625 V                                              |
| Linearidade Diferencial (DNL) | 0,60 LSB                                                           |
| Linearidade Integral (INL)    | 0,50 LSB                                                           |
| Máxima Frequência de Vin      | 1 MHz (Número Efetivo de Bits = 5,5 e $F_{\text{Clock}} = 20$ MHz) |
| Máxima Frequência de Clock    | 80 MHz (Número Efetivo de Bits = 5,5 e $F_{\text{Vin}} = 1$ KHz)   |
| Voltagem de Alimentação       | -5 V                                                               |
| Consumo de Potência           | 305 mW (medido) / 325 mW (simulado)                                |
| Transistores                  | 788                                                                |
| Resistores                    | 420                                                                |
| Capacitor                     | 1                                                                  |
| Código Binário de Saída       | Binário - ECL 10 K                                                 |

circuitos que trabalhem em alta frequência.

### 4.3 – Conversor A/D de 8 bits (topologia convencional):

#### 4.3.1 – Descrição:

A seguir serão apresentados os resultados medidos do conversor A/D de 8 bits com a topologia convencional e a proposta final de interpolação. Este conversor A/D não possui, porém, os circuitos de correção de erro e nem sincronismo de bits. O circuito integrado (CI) foi implementado em tecnologia BiCMOS de  $0,8 \mu\text{m}$ ,  $F_T$  de 12 GHz, p-substrato, duplo metal, duplo “polysilicon”, processo da AMS (Austria Mikro Systeme International AG). Este circuito integrado (FAPESP\_59) foi construído na rodada “A98-15” de 18 de setembro de 1998 da AMS, através do PMU / FAPESP e foi recebido em julho de 1999.

#### 4.3.2 – Micrografia do Circuito Integrado:

A Fig. 4.5 mostra a micrografia do circuito integrado do conversor A/D de 8 bits para um encapsulamento de 40 pinos (DIL). A área total é de  $1710,65 \mu\text{m} \times 4086,85 \mu\text{m} = 6,99 \text{ mm}^2$ . A área ficou maior do que o previsto em função da limitada experiência na elaboração de “layout”, o que causou dificuldade na compactação do circuito.

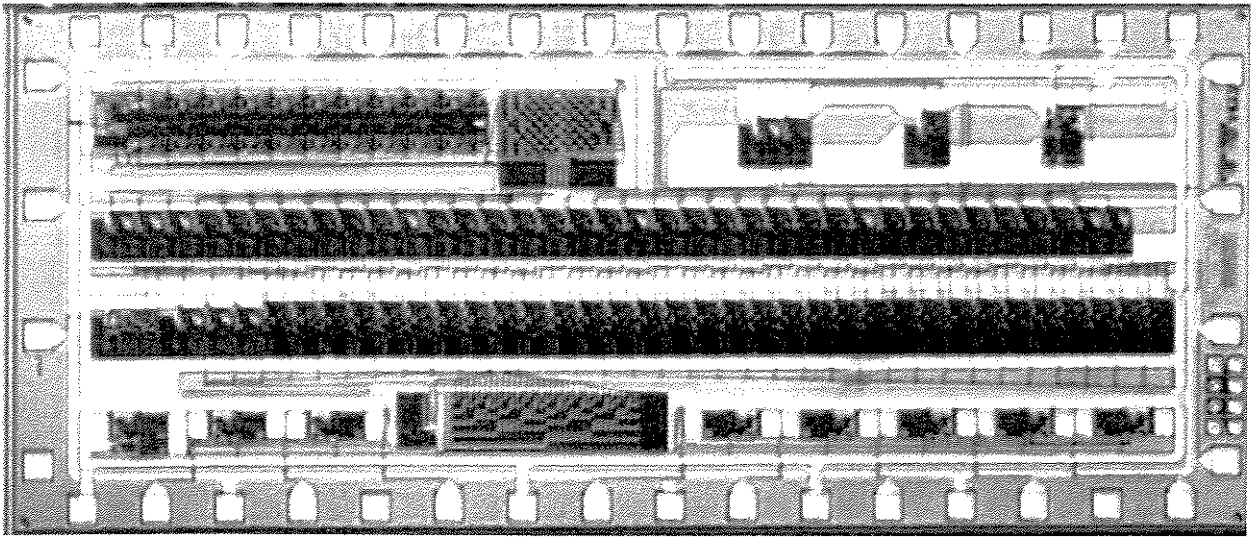


Figura 4.5 – Micrografia do Conversor A/D de 8 bits (Topologia Convencional)

#### 4.3.3 – Sistema de Teste:

A Fig. 4.6 mostra o sistema de teste aplicado, a Fig. 4.7 (a) e a Fig. 4.7 (b) mostram, respectivamente, o circuito e a placa usados nos testes e a Tabela 4.3 a relação dos equipamentos e software utilizados. O conversor D/A utilizado foi o HI20201JCP (Harris Semiconductor). As medidas realizadas tiveram por objetivo avaliar a performance estática (DNL e INL) e dinâmica (número efetivo de bits) do conversor A/D de 8 bits.

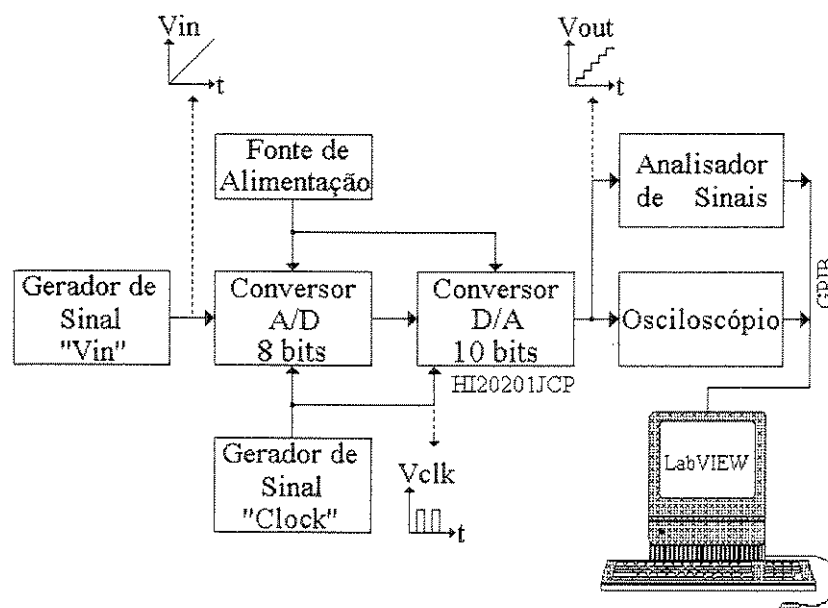


Figura 4.6 – Sistema de Teste

Tabela 4.3 – Relação dos Equipamentos e Software Utilizados

| Descrição            | Marca                | Modelo                       |
|----------------------|----------------------|------------------------------|
| Osciloscópio         | Hewlett Packard      | 54501A – 100 MHz             |
| Osciloscópio         | Hewlett Packard      | 54503A – 500 MHz             |
| Analizador de Sinais | Hewlett Packard      | 35660A – 100 KHz             |
| Gerador de Sinais    | Hewlett Packard      | 33120A – 15 MHz              |
| Gerador de Sinais    | Hewlett Packard      | 3314A – 20 MHz               |
| Gerador de Sinais    | Hungary              | TR-0614/B – 1172/B – 520 MHz |
| Fonte de Alimentação | Hewlett Packard      | E3631A                       |
| Aquisição de Dados   | National Instruments | LabVIEW 5.0                  |

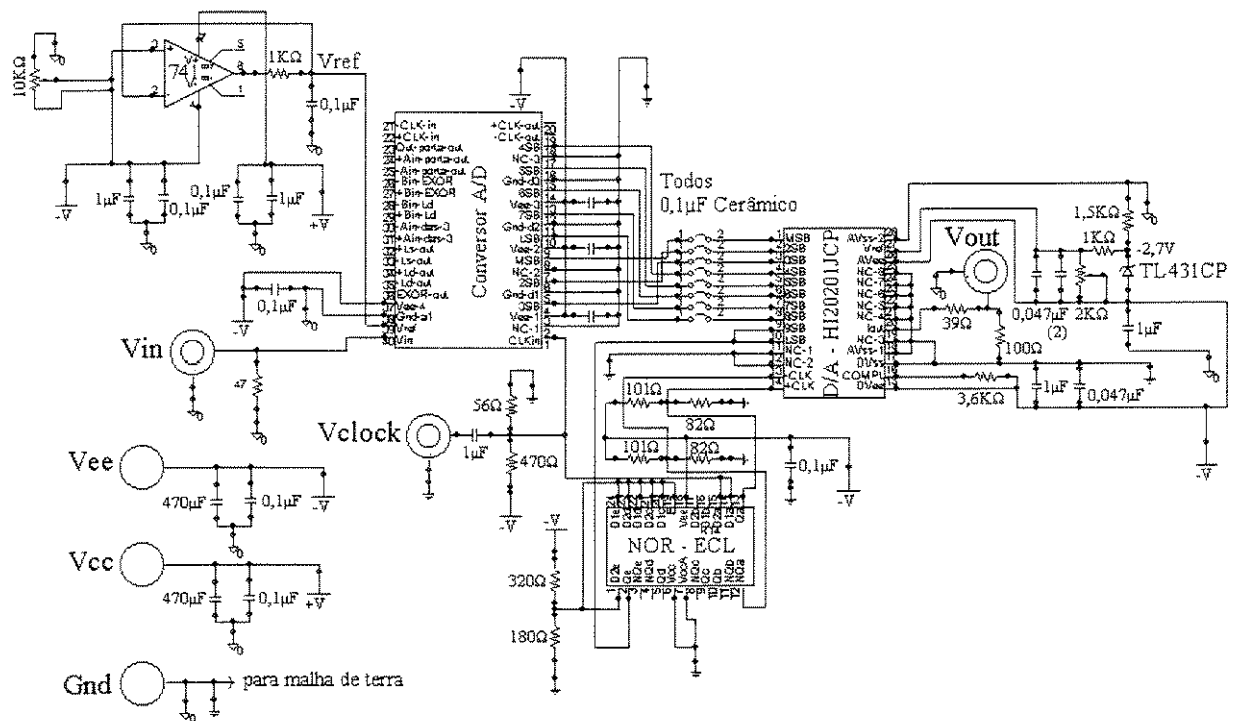


Figura 4.7 (a)– *Circuito da Placa Utilizada nos Testes*

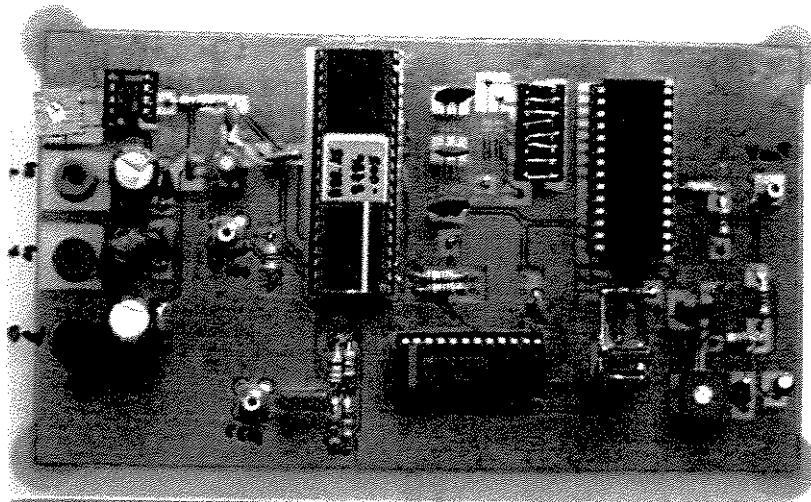


Figura 4.7 (b) – Placa Utilizada nos Testes

#### 4.3.4 – Consumo de Potência:

As fontes de polarização no conversor A/D utilizam resistores de “poly 2” (veja o item 3.12). De acordo com os parâmetros do processo, a resistência de folha típica para “poly 2” é de  $67 \Omega/\square$ , podendo chegar a um valor máximo de  $80 \Omega/\square$  e a um valor mínimo de  $55 \Omega/\square$ . As variações nos parâmetros de processo são portanto de +19,4% e de -17,9% respectivamente. Isto justifica a diferença no consumo de potência de ~ 8,96%.

Tabela 4.4 – Consumo de Potência

|                       |        |
|-----------------------|--------|
| Descrição/Temperatura | ~ 27°C |
| Consumo Simulado      | 748 mW |
| Consumo Medido        | 815 mW |

#### 4.3.5 – Sinal de Saída Reconstruído:

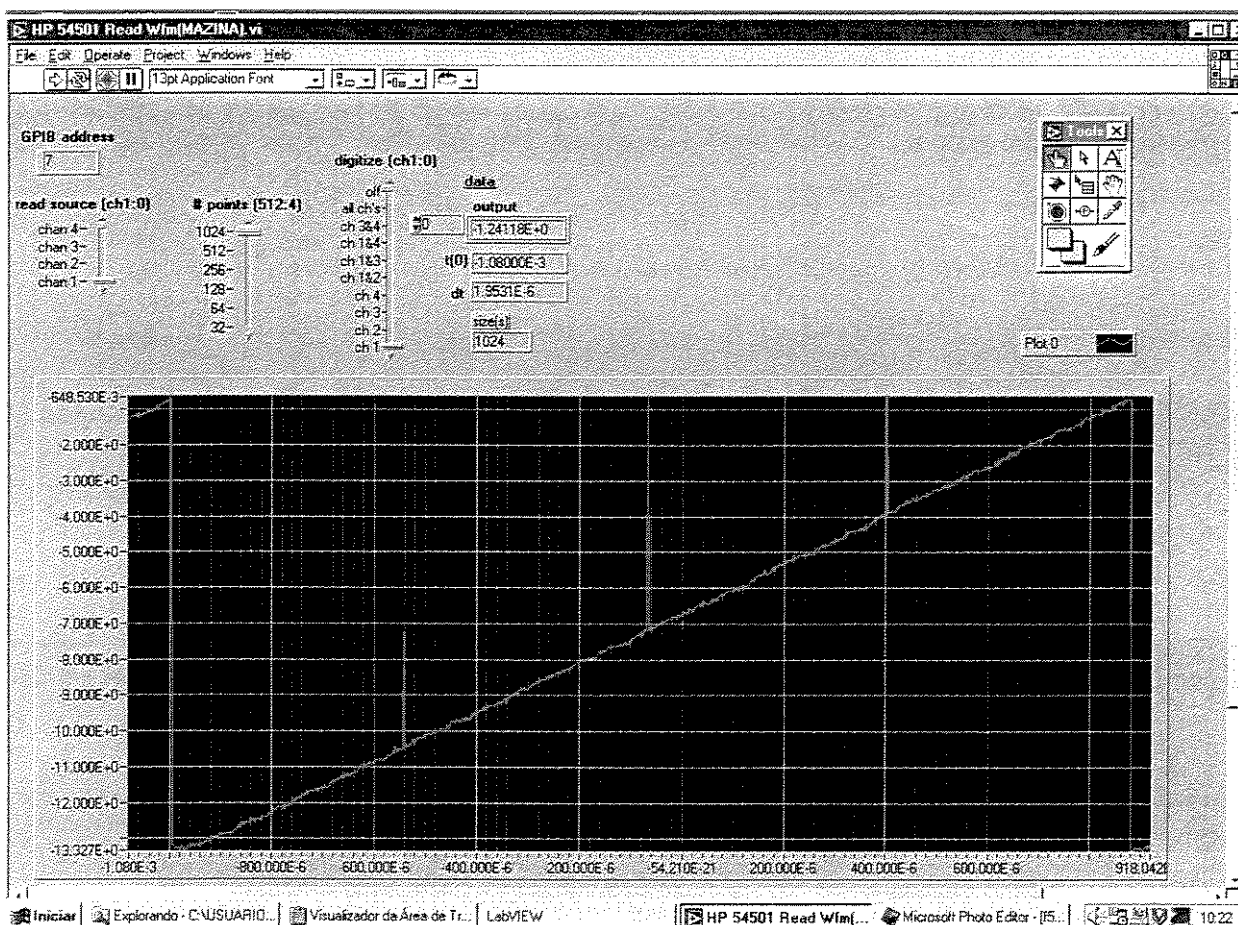


Figura 4.8 – Sinal de Saída Reconstruído

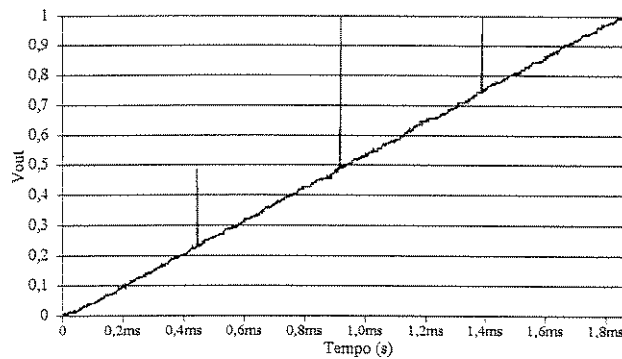


Figura 4.9 (a) – Sinal com “Glitches”

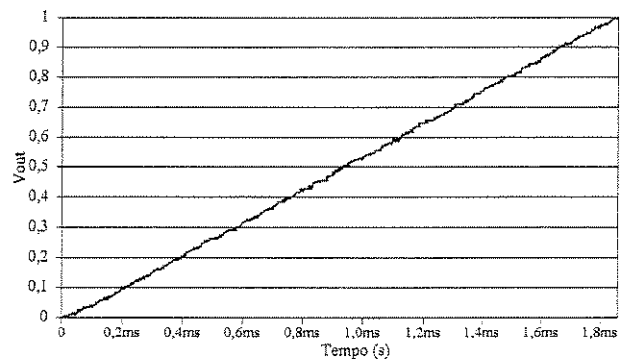


Figura 4.9 (b) – Sinal sem “Glitches”

A Fig. 4.8 mostra um exemplo do sinal de saída do conversor A/D de 8 bits reconstruído pelo conversor D/A para um sinal de entrada de 534 Hz e a frequência do sinal de “clock” de 2 MHz. Os “glitches” (transientes associados com a troca de código) presentes no sinal, são precisamente resultados da falta do circuito de sincronismo de bits entre o 3SB, o 2SB e o MSB. Estes “glitches” não se apresentam de forma constante no sinal reconstruído capturado pelo LabVIEW. Isto é, o sinal reconstruído capturado pode conter um, dois ou três “glitches” [veja a Fig. 4.9 (a)] ou, também, não conter nenhum [veja a Fig. 4.9 (b)] porque a existência dos “glitches” no sinal reconstruído depende do instante em que o sinal foi capturado. A frequência dos “glitches” depende do ajuste de temporização entre os “latches” dos bits mais significativos e varia com a frequência do sinal de “clock”.

#### 4.3.6 – Teste da Frequência de Batimento:

O teste da frequência de batimento (“beat frequency test”) [26] foi realizado colocando-se a frequência do sinal senoidal de entrada ( $f_{in} = 100$  KHz) um pouco diferente da frequência de amostragem ( $f_s = 101$  KHz). Uma diferença  $d_f = 1$  KHz, sendo  $d_f \ll f_{in}$ . Isto significa que o conversor A/D fará uma amostragem por período do sinal de entrada (veja a Fig. 4.10). Como a frequência do sinal de entrada é um pouco menor do que a frequência de amostragem, o instante amostrado irá deslocar um pouco a cada amostragem. De fato, o sinal de entrada será efetivamente amostrado a uma razão igual a  $f_s/d_f = 101$  [26]. O resultado é que o sinal de saída

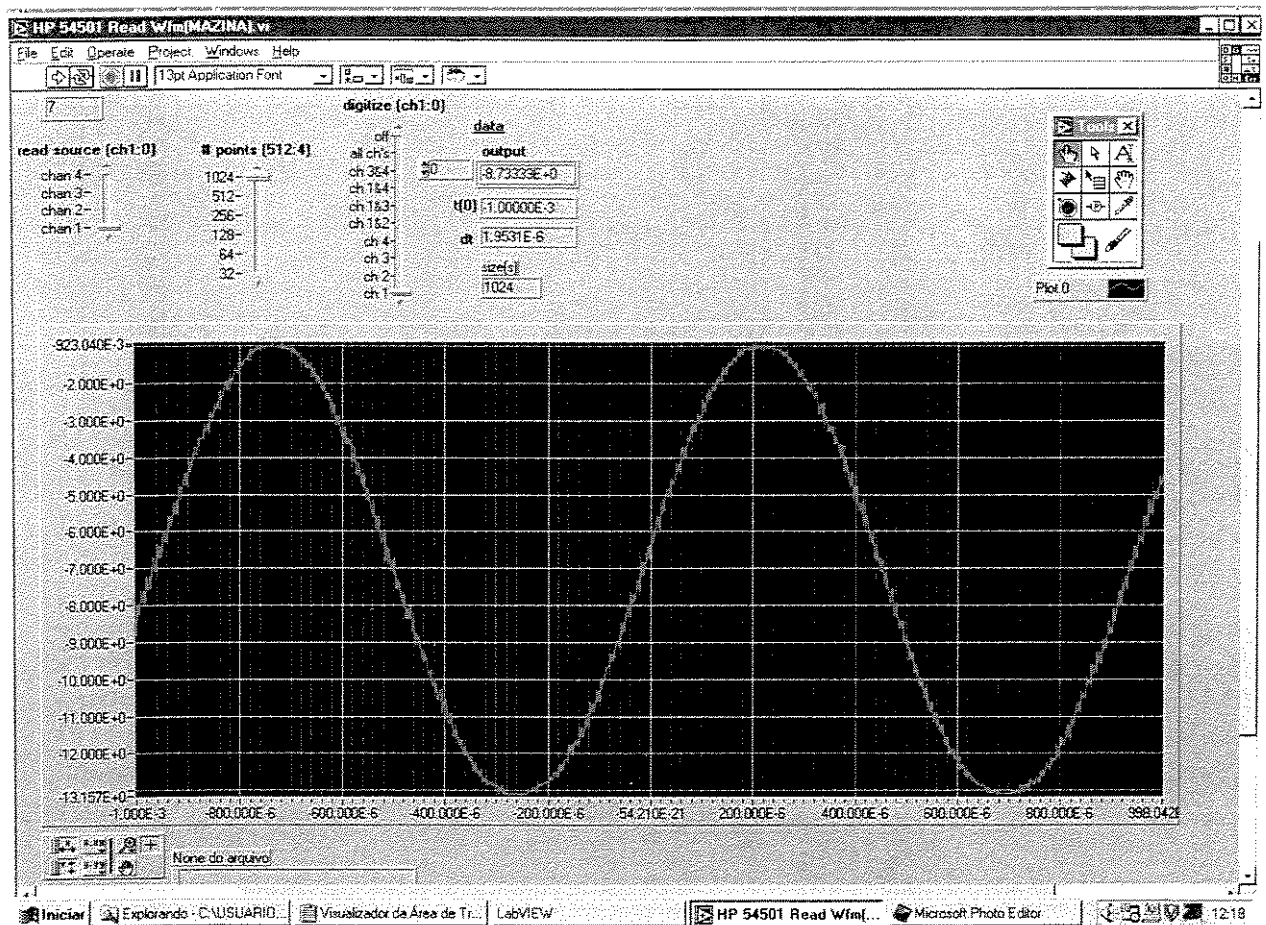


Figura 4.10 – *Teste da Frequência de Batimento*

reconstruído é senoidal com a frequência igual a  $d_f = 1$  KHz (chamada de “beat frequency”).

Este sinal senoidal com as suas imperfeições permite analisar qualitativamente o comportamento do conversor A/D. Principalmente a necessidade de sincronização dos três bits mais significativos.

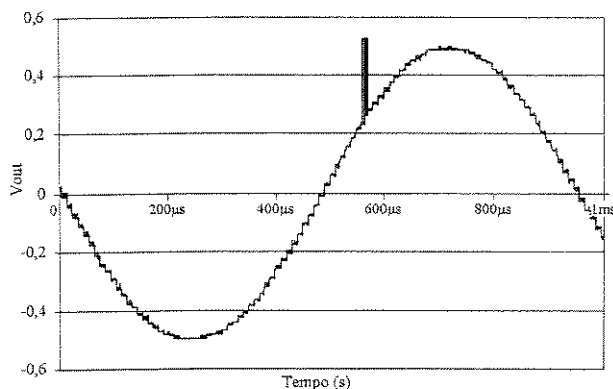


Figura 4.11 (a) – *Sinal com “Glitches”*

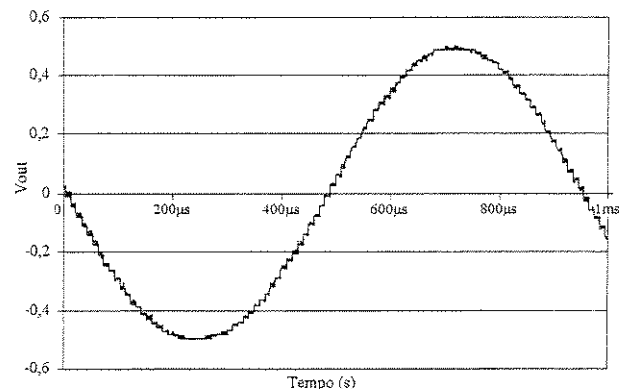


Figura 4.11 (b) – *Sinal sem “Glitches”*

De forma semelhante ao item anterior, os "glitches" presentes no sinal são resultados da falta do bit de sincronismo entre o 3SB, o 2SB e o MSB e não se apresentam de forma constante [veja as Fig. 4.11 (a) e Fig. 4.11 (b)]. Os motivos são semelhantes aos apresentados no item anterior.

#### 4.3.7 – Erro de Quantização:

O erro de quantização se apresentou bastante alto em determinados pontos, por dois motivos. O motivo principal é que a linearidade do conversor A/D ficou ruim em alguns pontos do sinal reconstruído. O motivo secundário é que todos os pequenos ruídos e “glitches” presentes no sinal aparecem na curva de quantização.

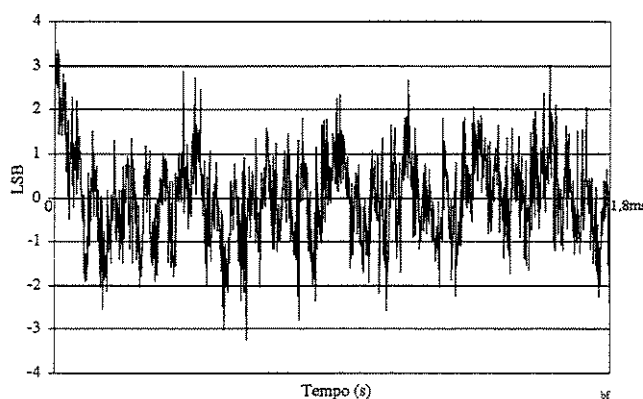


Figura 4.12 – Erro de Quantização

#### 4.3.8 – Resultados Estáticos:

As Fig. 4.13 (a) e 4.13 (b) mostram o erro de não linearidade diferencial (DNL) e o erro

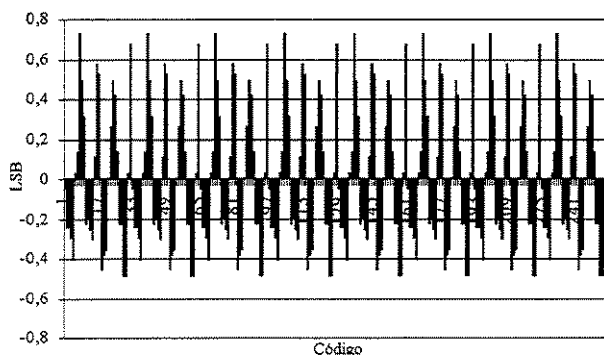


Figura 4.13 (a) – DNL

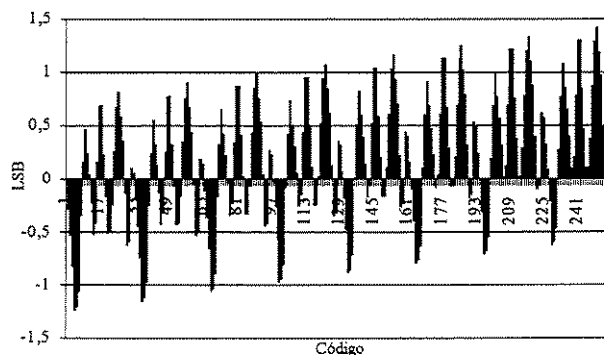


Figura 4.13 (b) – INL

Tabela 4.5 – *Resultados Estáticos*

| Medidas/Temperatura               | ~ 27°C   |
|-----------------------------------|----------|
| Não Linearidade Diferencial - DNL | 0,73 LSB |
| Não Linearidade Integral - INL    | 0,70 LSB |

de não linearidade integral (INL) respectivamente. A Tabela 4.5 mostra o resumo destes valores, considerando o maior valor encontrado do DNL e o valor do INL para o último degrau (em particular) do sinal reconstruído [25]. Os valores de linearidade (DNL e INL) observados e medidos em diversos artigos publicados para conversores A/D estão na faixa de: 0,75 LSB em [02, 11, 12 e 30] e 0,5 LSB em [06, 07, 08 e 09]. Contudo, o valor do INL apresentou um valor máximo de 1,4 LSB, o que pode ser considerado elevado (o ideal é ficar menor do que 1 LSB).

4.3.9 – *Resultados Dinâmicos:*

As Fig. 4.14 (a) e 4.14 (b) mostram o número efetivo de bits (NEB) em função da frequência do sinal de entrada ( $F_{vin}$ ) e em função da frequência do sinal de “clock” ( $F_{Clock}$ ) respectivamente. A Tabela 4.6 mostra um resumo dos valores medidos. A medida do número efetivo de bits em função da frequência do sinal de “clock” foi obtida através da análise da “Fast Fourier Transform - FFT”, realizada com o instrumento HP 35660A (100 KHz). O número

Tabela 4.6 – *Resultados Dinâmicos*

|                                       |                                              |
|---------------------------------------|----------------------------------------------|
| Máxima Frequência do Sinal de Entrada | ~ 2 MHz (para Número Efetivo de Bits = 7,5)  |
| Máxima Frequência do Sinal de “Clock” | ~ 45 MHz (para Número Efetivo de Bits = 7,5) |

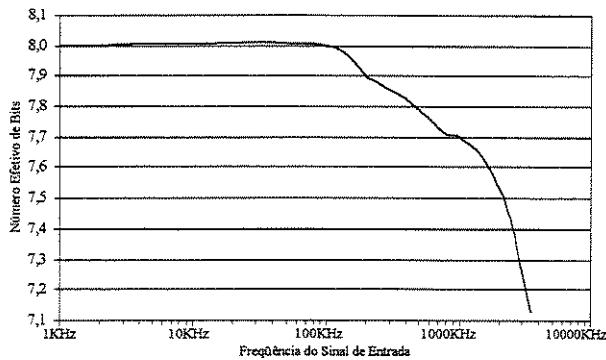


Figura 4.14 (a) – NEB em Função de  $F_{vin}$

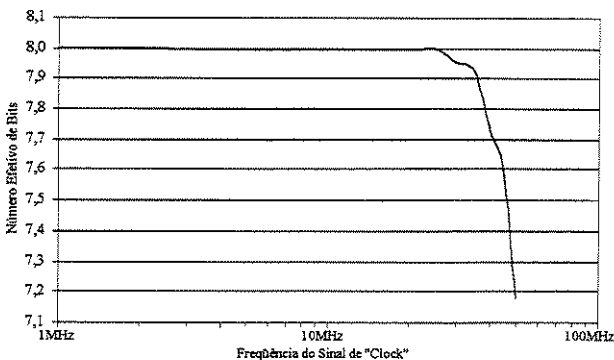


Figura 4.14 (b) – NEB em Função de  $F_{Clock}$

efetivo de bits em função da frequência do sinal de entrada também foi obtido através da análise da “Fast Fourier Transform - FFT”; porém, esta análise foi realizada por “software” (Origin 5.0) utilizando-se os sinais obtidos com o LabVIEW. A performance dinâmica ficou limitada pelos parasitas do “layout” e distribuição do sinal de “clock”. Contudo, a performance dinâmica pode ser melhorada adotando-se alguns cuidados especiais no “layout” e na distribuição do sinal de “clock” internamente no conversor A/D (por exemplo: usando-se “buffers” para o sinal de “clock” para cada grupo de “latches”).

#### 4.3.10 – Análise dos Cinco Bits Menos Significativos:

O padrão do sinal de saída reconstruído tende a se repetir em ciclos que correspondem ao comportamento dos cinco bits menos significativos. Isto torna pertinente uma análise mais detalhada destes cinco últimos bits. A Fig. 4.15 mostra os trinta e dois níveis (ou degraus) do sinal reconstruído, referentes aos cinco bits menos significativos. As Fig. 4.16 (a) e 4.16 (b) mostram o erro de não linearidade diferencial (DNL) e o erro de não linearidade integral (INL)

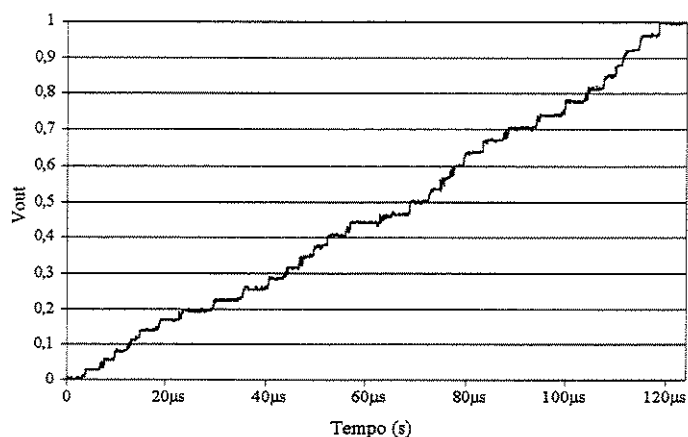


Figura 4.15 – Sinal Reconstruído dos Cinco Bits Menos Significativos

Tabela 4.7 – Resultados Estáticos (LSB's)

| Medidas/Temperatura               | ~ 27°C   |
|-----------------------------------|----------|
| Não Linearidade Diferencial - DNL | 0,73 LSB |
| Não Linearidade Integral - INL    | 0,08 LSB |

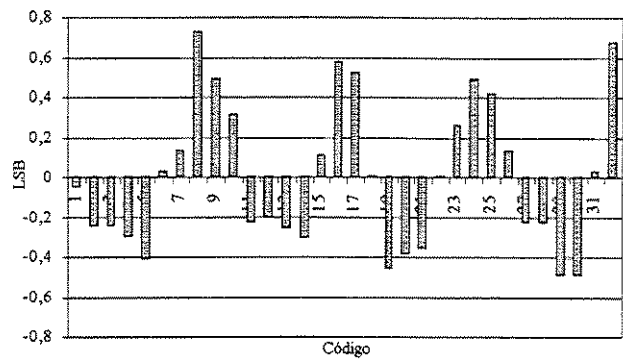


Figura 4.16 (a) – DNL

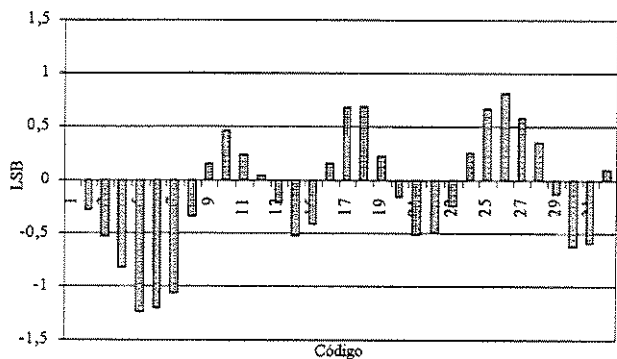


Figura 4.16 (b) – INL

respectivamente. A Tabela 4.7 mostra o resumo destes valores, considerando o maior valor encontrado do DNL e o valor do INL para o último degrau (em particular) do sinal reconstruído [25]. O valor máximo do INL foi de -1,2 LSB, quando o ideal seria ficar abaixo de 1 LSB.

4.3.11 – Especificações Gerais do Conversor A/D de 8 Bits:

A Tabela 4.8 mostra um resumo das características e dos resultados medidos na avaliação do conversor A/D de 8 bits com topologia convencional.

Tabela 4.8 – Especificações Gerais do Conversor A/D de 8 Bits

|                               |                                                             |
|-------------------------------|-------------------------------------------------------------|
| Resolução                     | 8 bits                                                      |
| Sinal de Entrada              | -1,0625 V a -0,0625 V                                       |
| Linearidade Diferencial (DNL) | 0,73 LSB                                                    |
| Linearidade Integral (INL)    | 0,70 LSB                                                    |
| Máxima Frequência de Vin      | 2 MHz (Número Efetivo de Bits = 7,5 e $F_{Clock} = 20$ MHz) |
| Máxima Frequência de Clock    | 45 MHz (Número Efetivo de Bits = 7,5 e $F_{Vin} = 1$ KHz)   |
| Voltagem de Alimentação       | -5 V                                                        |
| Consumo de Potência           | 815 mW (medido) / 748 mW (simulado)                         |
| Transistores                  | 2777                                                        |
| Resistores                    | 1325                                                        |
| Código Binário de Saída       | Binário - ECL 10 K                                          |

4.4 – Conversor A/D de 8 bits (com a nova topologia):

4.4.1 – Descrição:

A seguir, serão apresentados os resultados medidos do conversor A/D de 8 bits com a nova topologia, com a proposta final de interpolação, com circuitos de correção de erro e

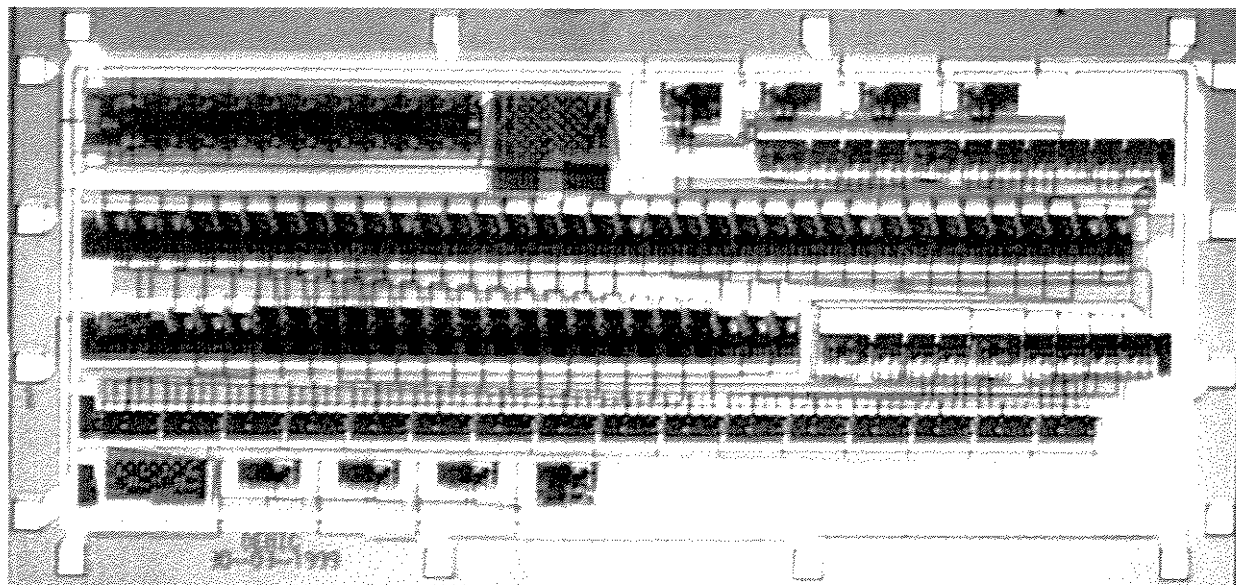


Figura 4.17 – *Micrografia do Conversor A/D de 8 bits (Nova Topologia)*

sincronismo de bits. O circuito integrado (CI) foi implementado em tecnologia BiCMOS de 0,8  $\mu\text{m}$ ,  $F_T$  de 12 GHz, p-substrato, duplo metal, duplo “polysilicon”, processo da AMS (Austria Mickro Systeme International AG). Este circuito integrado (FAPESP\_67) foi construído na rodada “A99-5” de 9 de abril de 1999 da AMS, através do PMU / FAPESP e foi recebido em setembro de 1999.

Devido a erro no “layout”, na parte de sincronismo de bits, os três primeiro bits do conversor A/D não funcionaram (MSB, 2SB e 3SB). Contudo, os cinco últimos bits que são resultados da interpolação, possuindo a correção de erro e sendo gerados com a nova topologia do circuito, funcionaram e somente estes foram objetos de avaliação.

#### 4.4.2 – *Micrografia do Circuito Integrado:*

A Fig. 4.17 mostra a micrografia do circuito integrado do conversor A/D de 8 bits para um encapsulamento de 16 pinos (DIL). A área total é de  $1876,05 \mu\text{m} \times 4086,95 \mu\text{m} = 7,67 \text{ mm}^2$ . A área ficou maior do que o previsto em função da limitada experiência na elaboração de “layout”, o que causou dificuldade na compactação do circuito. Apesar disto, a área ficou apenas

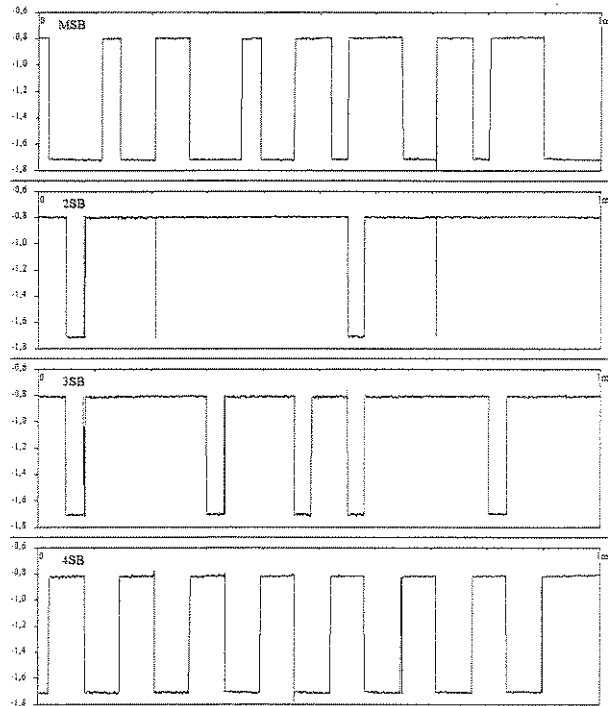


Figura 4.18 (a) – Saída Medida (real)

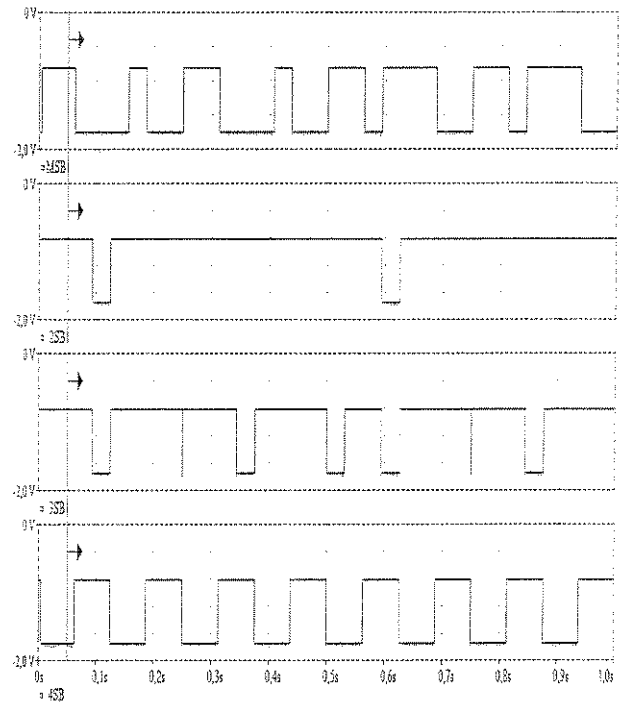


Figura 4.18 (b) – Saída Simulada

um pouco maior do que a área do conversor A/D de 8 bits ( $6,99 \text{ mm}^2$ ) com topologia convencional, não obstante ter sido incluída a correção de erro e o sincronismo de bits.

#### 4.4.3 – Erro no “Layout”:

O erro no “layout” foi a troca dos sinais de controle do sincronismo de bits, conforme pode ser visto na Tabela 4.9. Para melhor entendimento, a Fig. 4.18 (a) mostra o sinal de saída dos três primeiros bits do conversor A/D (MSB, 2SB e 3SB) que estão errados juntos com o quarto bit (4SB) que está correto. A Fig. 4.18 (b) mostra os mesmos sinais de saída do conversor A/D como resultado de uma simulação, onde foi aplicada propositalmente a mesma troca nos

Tabela 4.9 – Troca nos Sinais de Controle

| Sinal | Trocado Pelo     |
|-------|------------------|
| A     | $\overline{C}$   |
| B     | $\overline{B}$   |
| C     | $\overline{A}$   |
| AB    | $\overline{ABC}$ |
| ABC   | $\overline{AB}$  |

sinais de controle do sincronismo de bits. Este erro ocorreu porque as trilhas de ligação no “layout” foram nomeadas invertidas e apareceram no “netlist” como se estivessem corretas.

#### 4.4.4 – Sistema de Teste:

A Fig. 4.19 mostra o sistema de teste aplicado e a Tabela 4.10 a relação dos equipamentos e software utilizados. O conversor D/A utilizado foi o HI20201JCP (Harris Semiconductor). As medidas realizadas tiveram por objetivo avaliar a performance estática (DNL e INL) do conversor A/D.

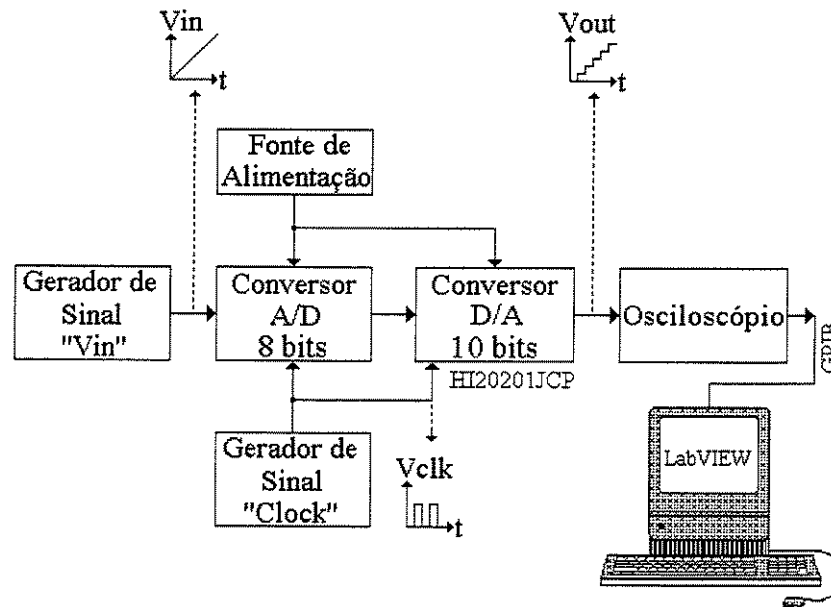


Figura 4.19 – Sistema de Teste

Tabela 4.10 – Relação dos Equipamentos e Software Utilizados

| Descrição            | Marca                | Modelo           |
|----------------------|----------------------|------------------|
| Osciloscópio         | Hewlett Packard      | 54503A – 500 MHz |
| Osciloscópio         | Hewlett Packard      | 54501A – 100 MHz |
| Gerador de Sinais    | Hewlett Packard      | 33120A – 15 MHz  |
| Gerador de Sinais    | Hewlett Packard      | 3314A – 20 MHz   |
| Fonte de Alimentação | Hewlett Packard      | E3631A           |
| Aquisição de Dados   | National Instruments | LabVIEW 5.0      |

4.4.5 – Consumo de Potência:

As fontes de polarização no conversor A/D utilizam resistores de “poly 2”. O processo utilizado neste conversor A/D é o mesmo utilizado no conversor A/D com topologia convencional (item 4.3.4). Desta forma, a explicação contida no item 4.3.4 também é válida para este conversor A/D. O que justifica a diferença no consumo de potência de ~ 10,13%.

Tabela 4.11 – Consumo de Potência

|                       |        |
|-----------------------|--------|
| Descrição/Temperatura | ~ 27°C |
| Consumo Simulado      | 612 mW |
| Consumo Medido        | 550 mW |

4.4.6 – Sinal de Saída Reconstituído:

A Fig. 4.20 (a) mostra o sinal de saída reconstituído no conversor D/A para um sinal de entrada (tipo triangular) aplicado ao conversor A/D. Em função da ausência dos três primeiros bits, verificam-se oito repetições dos cinco últimos bits, que são mostradas com maior detalhe na Fig. 4.20 (b).

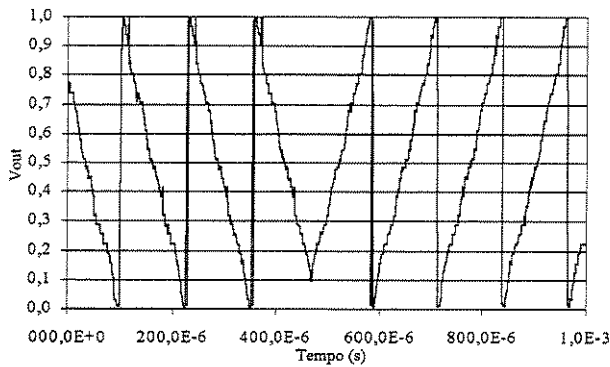


Figura 4.20 (a) – Sinal de Saída Reconstituído

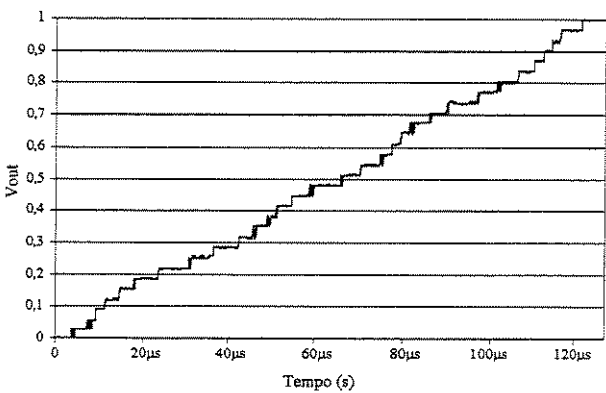


Figura 4.20 (b) – Cinco Últimos Bits

4.4.7 – Erro de Quantização:

A Fig. 4.21 mostra o erro de quantização. Pela ausência dos três primeiros bits, o erro de quantização foi considerado apenas para os cinco últimos bits do conversor A/D. O erro de quantização se apresentou bastante alto em alguns pontos, por dois motivos. O motivo principal é

que a linearidade do conversor A/D ficou ruim em alguns pontos do sinal reconstruído. O motivo secundário é que todos os pequenos ruídos e “gliches” presentes no sinal aparecem na curva de quantização.

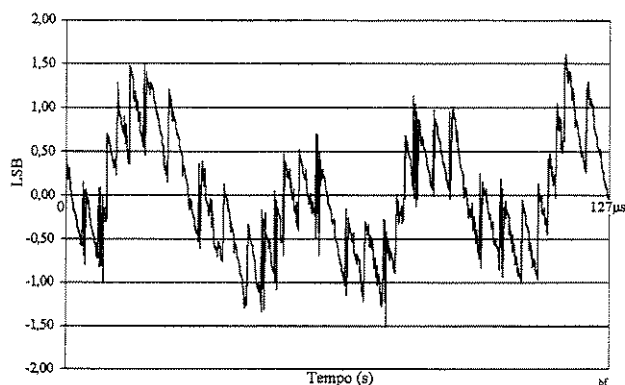


Figura 4.21 – Erro de Quantização

#### 4.4.8 – Resultados Estáticos:

Pela ausência dos três primeiros bits, a avaliação dos resultados estáticos foi considerada apenas para os cinco últimos bits do conversor A/D. As Fig. 4.22 (a) e 4.22 (b) mostram o erro de não linearidade diferencial (DNL) e o erro de não linearidade integral (INL) respectivamente. A Tabela 4.12 mostra o resumo destes valores, considerando o maior valor encontrado do DNL e o

Tabela 4.12 – Resultados Estáticos

| Medidas/Temperatura               | ~ 27°C   |
|-----------------------------------|----------|
| Não Linearidade Diferencial - DNL | 0,84 LSB |
| Não Linearidade Integral - INL    | 0,10 LSB |

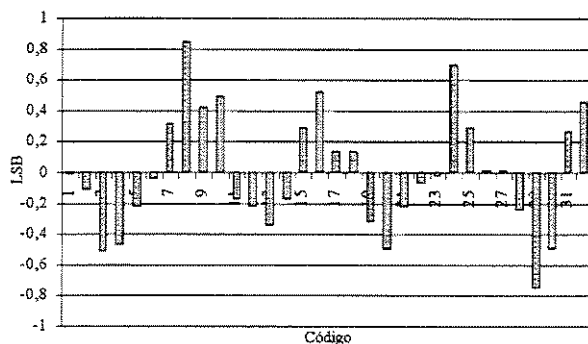


Figura 4.22 (a) – DNL

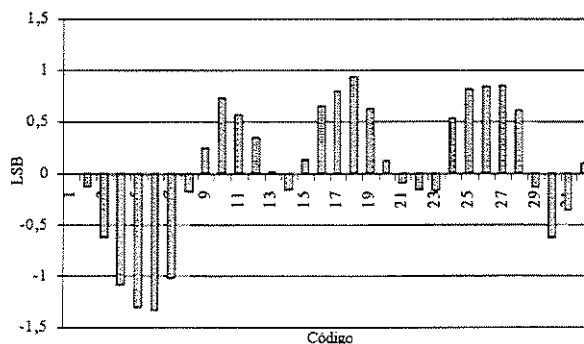


Figura 4.22 (b) – INL

valor do INL para o último degrau (em particular) do sinal reconstruído [25]. O valor do DNL ficou bastante alto, porém inferior a 1 LSB, o que não provocou perda de código. Contudo, o valor do INL apresentou um valor máximo de -1,4 LSB, quando o ideal seria ficar abaixo de 1 LSB.

#### 4.4.9 – Especificações Gerais do Conversor A/D de 8 Bits:

A Tabela 4.13 mostra um resumo das características e dos resultados medidos na avaliação do conversor A/D de 8 bits com a nova topologia.

Tabela 4.13 – Especificações Gerais do Conversor A/D de 8 Bits

|                               |                                                     |
|-------------------------------|-----------------------------------------------------|
| Resolução                     | 8 bits (apenas os 5 LSB's funcionando corretamente) |
| Sinal de Entrada              | -1,0625 V a -0,0625 V                               |
| Linearidade Diferencial (DNL) | 0,84 LSB (para os cinco últimos bits)               |
| Linearidade Integral (INL)    | 0,10 LSB (para os cinco últimos bits)               |
| Voltagem de Alimentação       | -5 V                                                |
| Consumo de Potência           | 550 mW (medido) / 612 mW (simulado)                 |
| Transistores                  | 2087                                                |
| Resistores                    | 588                                                 |
| Código Binário de Saída       | Binário - ECL 10 K                                  |

### 4.5 – Análise dos Resultados:

#### 4.5.1 – Interpolação:

O DNL e o INL do conversor A/D dependem principalmente da eficiência da interpolação. A Fig. 4.23 (a) mostra os sinais interpolados mais os sinais que fazem a interpolação ( $3SB/3SB$ ,  $S135/S135$ ,  $S90/S90$  e  $S45/S45$ ). A interpolação é realizada entre

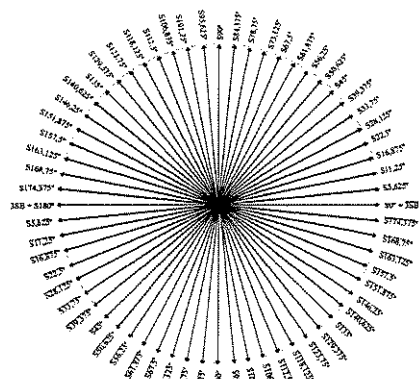


Figura 4.23 (a) – Sinais

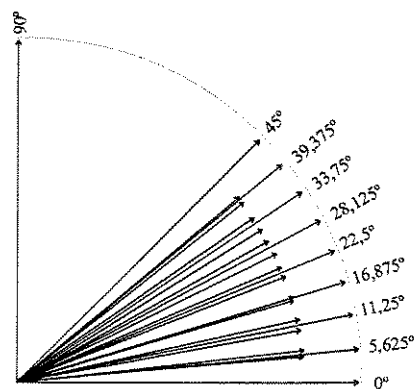


Figura 4.23 (b) – Interpolação

$3SB/\overline{3SB}$  e  $S135/\overline{S135}$ , entre  $S135/\overline{S135}$  e  $S90/\overline{S90}$ , entre  $S90/\overline{S90}$  e  $S45/\overline{S45}$ , e entre  $3SB/\overline{3SB}$  e  $S135/\overline{S135}$  (veja o capítulo 2). Cada uma destas interpolações segue o mesmo padrão, que é semelhante ao mostrado na Fig. 4.23 (b). Desta forma, este padrão se repete como pode ser percebido na Fig. 4.20 (b) e se aplica aos dois conversores A/D de 8 bits construídos que possuem a proposta final de interpolação com as mesmas relações de áreas.

Na simulação, as relações de áreas adotadas e ângulos resultantes se mostraram perfeitas. Na prática, houve uma diferença que provocou o aumento nos valores do DNL e INL. Os parâmetros do transistor para resistência de base e correntes de superfície variam em função da sua área e da sua geometria. Esta variação é considerada linear pelo programa de simulação, o que pode não corresponder na prática. Por exemplo: dois transistores (área = 1) em paralelo têm uma resistência de base resultante dividida por dois. Um transistor com duas vezes a área (área = 2) tem, na prática, uma resistência de base resultante um pouco diferente daquela dividida por dois e o programa de simulação irá considerar igual à resistência de base dividida por dois. Este fato justifica a diferença entre o resultado simulado e o resultado medido. A relação entre  $\Delta I_c/I_c$  em função de  $\Delta r_b$  para 2 transistores em paralelo (com o mesmo modelo, considerando apenas os  $r_b$  diferentes) pode ser estimada pela expressão a seguir:

$$\frac{\Delta I_c}{I_c} \cong \frac{\Delta r_b}{\beta} \frac{1}{\left(\frac{V_T}{I_c} - \frac{r_{b2}}{\beta}\right)}$$

A Fig. 4.24 mostra as curvas entre a razão de áreas e ângulos resultantes obtidos na simulação em comparação com os resultados medidos na prática (média dos dois conversores A/D de 8 bits). Observando as Fig. 4.15 e 4.20 (b), é possível perceber uma semelhança entre as curvas e entre os valores de DNL e INL (Fig. 4.15 e 4.21) para os dois conversores A/D de 8 bits. O que permite inferir que o aumento dos valores do DNL e INL ocorreu em função das relações

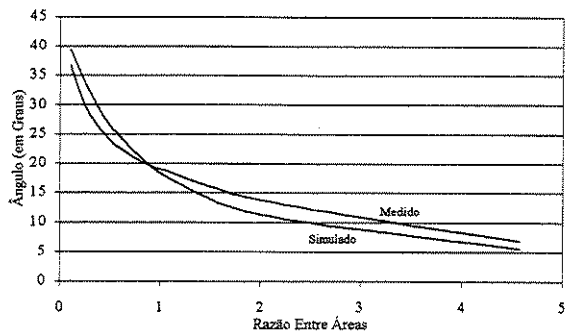


Figura 4.24 – *Interpolação: Razão de Áreas e Ângulos Resultantes*

de áreas adotadas para transistores e não pelo descasamento entre os transistores. As relações de áreas podem ser corrigidas, melhorando o

DNL e o INL do conversor A/D através da implementação de um circuito integrado para teste entre as diversas relações de áreas.

4.5.2 – Correção de Erro de Código:

A técnica de correção de erro melhora a performance em alta frequência do conversor A/D porque melhora a razão de erro. A correção de erro não pôde ser avaliada porque o conversor A/D de 8 bits com correção de erro, não teve uma análise dinâmica (pela ausência dos três bits mais significativos). De forma geral, o sinal reconstruído, que foi observado na saída, não apresentou nenhuma evidência de perda de bit ou “glitch” que pudesse ter sido causada por erro no código circular interno do conversor A/D.

4.5.3 – Nova Topologia:

A nova topologia se mostrou adequada, principalmente porque houve uma significativa redução do consumo no conversor A/D de 8 bits (com a nova topologia) em relação ao conversor A/D de 8 bits (com a topologia convencional), apesar da inclusão da correção de erro e do sincronismo de bits. A Tabela 4.14 ilustra bem as vantagens da nova topologia.

Tabela 4.14 – *Comparação Entre as Topologias*

| Descrição           | Topologia Convencional | Nova Topologia |
|---------------------|------------------------|----------------|
| Interpolação        | Sim                    | Sim            |
| Correção de Erro    | Não                    | Sim            |
| Sincronismo de Bits | Não                    | Sim            |
| Consumo de Potência | 815 mW                 | 550 mW         |
| Transistores        | 2777                   | 2087           |
| Resistores          | 1325                   | 588            |

## Conclusões Finais

Este trabalho propôs aproveitar melhor a arquitetura básica dos conversores A/D tipo duplo “folding” e realizar a interpolação no “folding encoder” e nos próprios “latches” mestres do conversor A/D. A vantagem é que a nova técnica de interpolação evita a rede de interpolação resistiva e alguns dos seus problemas com praticamente a mesma eficiência no consumo de potência.

Além disso, este trabalho apresentou um circuito com pequena complexidade, simples implementação e que corrige a maioria e os mais frequentes erros do código circular interno do conversor A/D. Apesar da correção de erro não ter sido avaliada na prática (em função do erro no “layout”), verificou-se através da simulação que a técnica de correção de erro, a qual detecta e corrige erros de bolha durante a detecção da transição de zero para um do código circular, melhora a razão de erro do conversor A/D e a sua performance em alta frequência.

Para avaliação das propostas, foram projetados, simulados e implementados três conversores A/D usando-se a arquitetura duplo “folding”: um conversor A/D de 6 bits com a proposta inicial de interpolação, um conversor A/D de 8 bits com topologia convencional e com a proposta final de interpolação e um conversor A/D de 8 bits com uma nova topologia de circuito, a proposta final de interpolação, a nova técnica de correção de erro e o sincronismo de bits.

Foi demonstrado que a proposta final de interpolação descrita provê outra alternativa com respeito à interpolação resistiva, acrescentando alguns transistores no circuito do “folding encoder” e nos “latches” mestres de interpolação. Os resultados obtidos mostraram que a nova técnica de interpolação torna possível um conversor A/D com resolução de 8 bits. Porém, deve-se considerar um cuidado especial nas áreas dos transistores de interpolação (nos “latches” de interpolação) porque a linearidade (DNL e INL) dependerá quase que exclusivamente da precisão

das razões das áreas.

Adicionalmente, para o bom desempenho do conversor A/D é importante implementar a correção de erro de código e o sincronismo de bits e aplicar cuidados adicionais na elaboração do “layout” do circuito e na distribuição do sinal de “clock”. Sob estas condições, o conversor A/D implementado deve alcançar resultados próximos aos obtidos na simulação. A nova topologia utilizada para o circuito demonstrou que permite a diminuição do número total de transistores e a correspondente diminuição no consumo total de potência.

Diante dos resultados obtidos, conclui-se que as inovações apresentadas possibilitam a construção de conversores A/D com resolução de até 8 bits e com a expectativa de ter um desempenho dinâmico similar a outros conversores A/D tipo “folding” com interpolação apresentados, porém, com menor área e consumo de potência (considerando a mesma tecnologia). Como continuidade desse trabalho, será construído outro conversor A/D de 8 bits com a correção no erro de “layout” (na parte de sincronismo de bits, veja o item 4.4.3), com a entrada do sinal de “clock” diferencial e uma melhor distribuição interna deste sinal e, ainda, cuidados adicionais no “layout” (no “placement” e no roteamento).

Numa etapa posterior, o que pode ser considerado para a continuidade e o aprimoramento desta proposta é a conversão do circuito para a tecnologia CMOS. Nesta conversão de tecnologia, poderiam ser aproveitadas a parte da chamada interpolação simples, a técnica de correção de erro (que poderia ser melhorada e corrigir os eventuais meta-estados do “latches”) e a topologia do circuito. Apenas a parte da interpolação dupla iria necessitar de uma completa alteração.

## Referência Bibliográfica

- [01]-*A. Abel e K. Kurtz* - "Fast ADC" - IEEE Trans. Nucl. Sci., Vol. NS-22, Fevereiro 1975, pgs. 446-451.
- [02]-*Udo Fiedler e Dieter Seitzer* - "A High Speed 8 Bit A/D Converter Based on a Gray-Code Multiple Folding Circuit" - IEEE Journal of Solid State Circuits, Vol. SC-14, No. 3, Junho 1979, pgs. 547-551.
- [03]-*Rudy J. van de Plassche e Rob E. J. van de Grift* - "A High-Speed 7 Bit A/D Converter" - IEEE Journal of Solid State Circuits, Vol. SC-14, No. 6, Dezembro 1979, pgs. 938-943.
- [04]-*Rob E. J. van de Grift e Rudy J. van de Plassche* - "A Monolithic 8-Bit Video A/D Converter" - IEEE Journal of Solid State Circuits, Vol. SC-19, No. 3, Junho 1984, pgs. 374-378.
- [05]-*Rob E. J. Van de Grift, Ivo W. J. M. Rutten e Martien Van der Veen* - "An 8 bit Video ADC Incorporating Folding and Interpolation Techniques" - IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 6, Dezembro 1987, pgs. 944 a 953.
- [06]-*Rudy J. van de Plassche e Peter Baltus* - "An 8-bit 100-MHz Full-Nyquist Analog-to-Digital Converter" - IEEE Journal of Solid-State Circuits, Vol. 23, No. 6, Dezembro 1988, pgs. 1334 a 1344.
- [07]-*Johan van Valburg e Rudy J. van de Plassche* - "An 8-b 650-MHz Folding ADC" - IEEE Journal of Solid-State Circuits, Vol. 27, No. 12, Dezembro 1992, pgs. 1662-1666.
- [08]-*Rudy J. van de Plassche* - "Integrated Analog to Digital and Digital to Analog Converters" - 1ª ed., Netherlands, Kluwer Academic Publishers, 1994, 501 pgs.
- [09]-*Bram Nauta e Ardie G. W. Venes* - "A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. 30, No. 12, Dezembro 1995, pgs. 1302-1308.
- [10]-*C. K. Poulton, K. L. Knudsen e J. J. Corcoran* - "A 6b 4 GS/s GaAs HBT ADC"- IEEE Journal of Solid State Circuits, Vol. 30, No. 10, Outubro 1995, pgs. 1109-1118.
- [11]-*Michael P. Flynn e David J. Allstot* - "CMOS Folding A/D Converters with Current-Mode Interpolation"- IEEE Journal of Solid State Circuits, Vol. 31, No. 9, Setembro 1996, pgs. 1248-1257.
- [12]-*Ardie G. W. Venes e Rudy J. van de Plassche* - "An 80-MHz, 80-mW, 8-b CMOS Folding A/D Converter with Distributed Track-and-Hold Preprocessing" - IEEE Journal of Solid State Circuits, Vol. 31, No. 12, Dezembro 1996, pgs. 1846-1853.
- [13]-*R. Alini et al.* - "A 200 MSample/s Trellis-Coded PRML, Read/Write Channel with Analog Adaptive Equalizer and Digital Servo" - IEEE Journal of Solid State Circuits, Vol. 32, No. 11, Novembro 1997, pgs. 1824-1838.
- [14]-*Pieter Vorenkamp e Raf Roovers* - "A 12-b, 60-MSample/s Cascaded Folding and Interpolating ADC" - IEEE Journal of Solid State Circuits, Vol. 32, No. 12, Dezembro 1997, pgs. 1876-1886.
- [15]-*K. Bult e A. Buchwald* - "An Embedded 240-mW 10-b 50-MS/s CMOS ADC in 1-mm<sup>2</sup>" - IEEE Journal of Solid State Circuits, Vol. 32, No. 12, Dezembro 1997, pgs. 1887-1895.

- [16]-*Michael P. Flynn e Ben Sheahan* - "A 400-MSample/s, 6-b CMOS Folding and Interpolating ADC" - IEEE Journal of Solid State Circuits, Vol. 33, No. 12, Dezembro 1998, pgs. 1932-1938.
- [17]-*Michael J. Demler*- "High-Speed Analog-To-Digital Conversion" - 1ª ed., United Kingdom, Academic Press, Inc., 1991, 218 pgs.
- [18]-*Christopher W. Mangelsdorf* - "A 400 MHz Input Flash Converter with Error Correction" - IEEE Journal of Solid-State Circuits, Vol. 25, No. 1, Fevereiro 1990, pgs. 184-191.
- [19]-*J. Peterson* - "A Monolithic Video A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. SC-14, No. 6, Dezembro 1979, pgs. 932-937.
- [20]-*Y. Akazama et al.* - "A 400 Msps 8b flash AD conversion LSI" – ISSCC Dig. Tech. Papers, Vol. 30, 1987, pgs. 98-99.
- [21]-*V. Garuts, Y. Yu, E. Traa and T. Yamaguchi* - "A dual 4-bit 2-Gs/s full Nyquist analog-to-digital converter using a 70-ps silicon bipolar technology with borosenic-poly process and coupling-base implant" - IEEE Journal of Solid-State Circuits, Vol. 24, No. 2, Abril 1989, pgs. 216-222.
- [22]-*Sanroku Tsukamoto, William G. Schofield and Toshiaki Endo* - "A CMOS 6-b, 400-Msample/s ADC with Error Correction" - IEEE Journal of Solid-State Circuits, Vol. 33, No. 12, Dezembro 1998, pgs. 1939-1947.
- [23]-*Eric A. Vittoz* - "The Design of High-Performance Analog Circuits on Digital CMOS Chips" - IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 3, Junho 1985, pgs. 657-665.
- [24]-*Simon J. Lovett, Marco Welten, Alan Mathewson e Barry Mason* - "Optimizing MOS Transistor Mismatch" - IEEE Journal of Solid-State Circuits, Vol. 33, No. 1, Janeiro 1998, pgs. 147-150.
- [25]-*Texas Instruments Incorporated* - "Understanding Data Converters" - 1ª ed., USA, Texas Instruments Incorporated, Julho 1995, 19 pgs.
- [26]-*Clyde F. Coombs* - "Electronic Instruments Handbook" - 2ª ed., USA, McGraw-Hill Inc., 1995, 1000 pgs.
- [27]-*Analog Devices* - "Analog-Digital Conversion Handbook", 3ª ed., New Jersey - USA, PTR Prentice Hall, 1969, 672 pgs.
- [28]-*Hiroshi Kimura; Akira Matsuzawa; Takashi Nakamura e Shigeaki Sawada*, - "A 10 b 300 Mhz Interpolated - Parallel A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. 28, No. 4, April 1993, pgs. 438 a 446.
- [29]-*Iuri Mehr e Declan Dalton* - "A 500-Msample/s, 6-Bit Nyquist-Rate ADC for Disk-Drive Read-Channel Applications"- IEEE Journal of Solid State Circuits, Vol. 34, No. 7, Julho 1999, pgs. 912-920.
- [30]- *Texas Instruments Incorporated* - "TLC5510, TLC5510A - 8-Bit High-Speed Analog-to-Digital Converters" - Texas Instruments Incorporated, September 1994 - Revised May 1999, 15 pgs. - Datasheet.

## Bibliografia Consultada

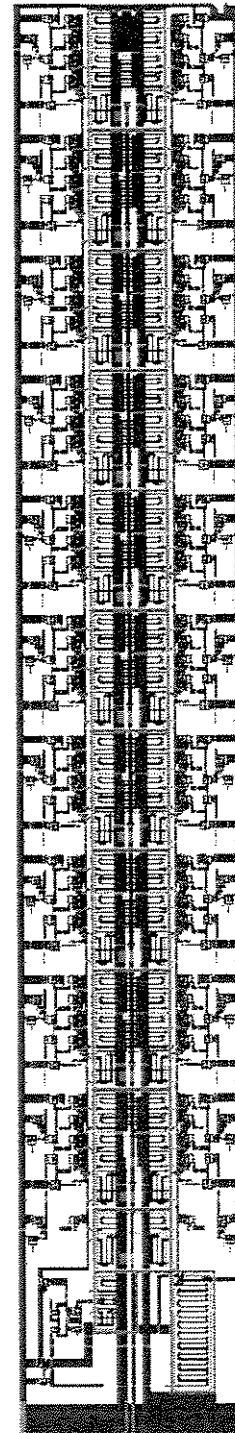
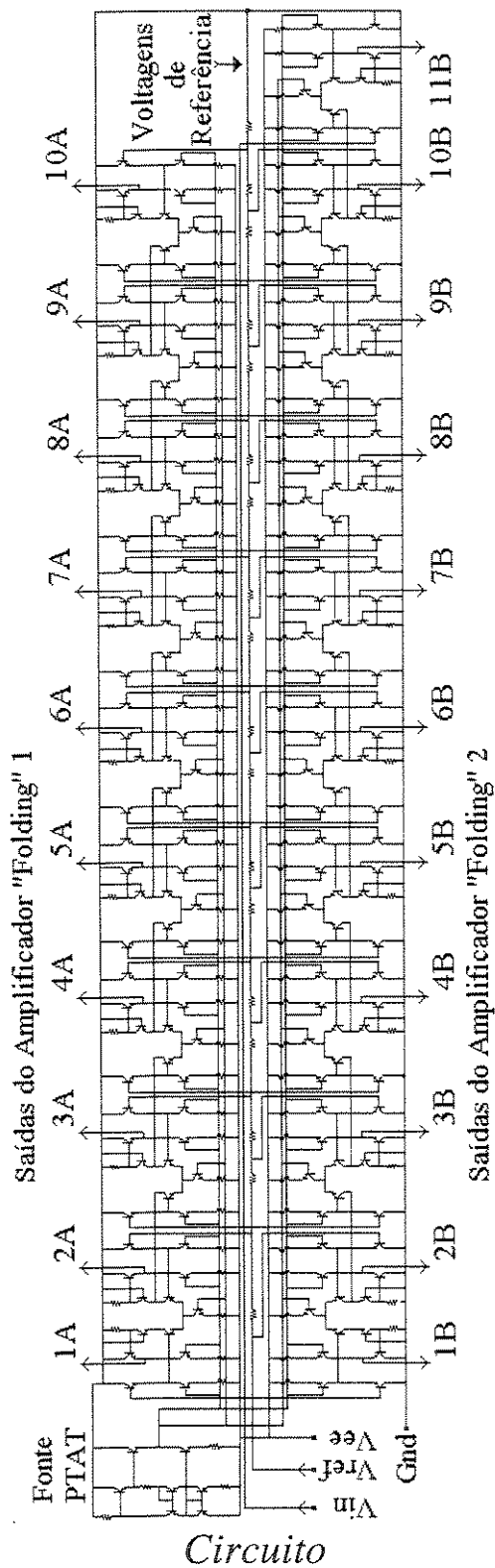
- *A. Abel e K. Kurtz* - "Fast ADC" - IEEE Trans. Nucl. Sci., Vol. NS-22, Fevereiro 1975, pgs. 446-451.
- *Alan B. Grebene* - "Bipolar and MOS Analog Integrated Circuit Design", 1ª ed., USA, John Wiley & Sons, Inc., 1984, 894 pgs.
- *Analog Devices* - "Analog-Digital Conversion Handbook", 3ª ed., New Jersey - USA, PTR Prentice Hall, 1969, 672 pgs.
- *Antônio Sérgio Cavalcanti de Menezes* - "Conversão Analógica-Digital Ultra-Rápida em Corrente em Tecnologia Bipolar - Nova Proposta" - Tese de Doutorado, UNICAMP, 1990, 109 pgs.
- *Ardie G. W. Venes e Rudy J. van de Plassche* - "An 80-MHz, 80-mW, 8-b CMOS Folding A/D Converter with Distributed Track-and-Hold Preprocessing" - IEEE Journal of Solid State Circuits, Vol. 31, No. 12, Dezembro 1996, pgs. 1846-1853.
- *Bram Nauta e Ardie G. W. Venes* - "A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. 30, No. 12, Dezembro 1995, pgs. 1302-1308.
- *Christopher W. Mangelsdorf* - "A 400 MHz Input Flash Converter with Error Correction" - IEEE Journal of Solid-State Circuits, Vol. 25, No. 1, Fevereiro 1990, pgs. 184-191.
- *Clyde F. Coombs* - "Electronic Instruments Handbook" - 2ª ed., USA, McGraw-Hill Inc., 1995, 1000 pgs.
- *C. K. Poulton, K. L. Knudsen e J. J. Corcoran* - "A 6b 4 GS/s GaAs HBT ADC" - IEEE Journal of Solid State Circuits, Vol. 30, No. 10, Outubro 1995, pgs. 1109-1118.
- *David F. Hoeschele Jr.* - "Analog-to-Digital/Digital-to-Analog Conversion Techniques", 1ª ed., USA, John Wiley & Sons, Inc, 1968, 455 pgs.
- *Eric A. Vittoz* - "The Design of High-Performance Analog Circuits on Digital CMOS Chips" - IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 3, Junho 1985, pgs. 657-665.
- *Evandro Mazina Martins* - "Projeto de Um Conversor Analógico/Digital Ultra-Rápido Bipolar Tipo Folding Com Uma Nova Técnica de Interpolação" - Dissertação de Mestrado, UNICAMP, 1996, 117 pgs.
- *Hermann Schmid* - "Electronic Analog/Digital Conversions", 1ª ed., USA, Van Nostrand Reinhold Company, 1970, 527 pgs.
- *Hiroshi Kimura, Akira Matsuzawa, Takashi Nakamura e Shigeaki Sawada*, - "A 10 b 300 Mhz Interpolated - Parallel A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. 28, No. 4, Abril 1993, pgs. 438 a 446.
- *Iuri Mehr e Declan Dalton* - "A 500-Msample/s, 6-Bit Nyquist-Rate ADC for Disk-Drive Read-Channel Applications" - IEEE Journal of Solid State Circuits, Vol. 34, No. 7, Julho 1999, pgs. 912-920.
- *Jeffrey Y. F. Tang e J. Leon Yang* - "Noise Issues in the ECL Circuit Family" - Western Research Laboratory, Janeiro 1990, 61 pgs.

- *Joey Doernberg; Paul R. Gray e David A. Hodges* - "A 10 bit 5 Msample/s CMOS Two-Step Flash ADC" - IEEE Journal of Solid-State Circuits, Vol. 24, No 2, Abril 1989, pgs. 241 a 249.
- *Johan van Valburg e Rudy J. van de Plassche* - "An 8-b 650-MHz Folding ADC" - IEEE Journal of Solid-State Circuits, Vol. 27, No. 12, Dezembro 1992, pgs. 1662-1666.
- *J. Peterson* - "A Monolithic Video A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. SC-14, No. 6, Dezembro 1979, pgs. 932-937.
- *K. Bult e A. Buchwald* - "An Embedded 240-mW 10-b 50-MS/s CMOS ADC in 1-mm<sup>2</sup> " - IEEE Journal of Solid State Circuits, Vol. 32, No. 12, Dezembro 1997, pgs. 1887-1895.
- *Kevin M. Daugherty* - "Analog to Digital Conversion – A Practical Approach", 1ª ed., USA, McGraw-Hill, Inc., 1995, 214 pgs.
- *K. L. Cattermole* - "Principles of Pulso Code Modulation", 1ª ed., Londres, Iliffe Books Ltd, 1969, 447 pgs.
- *Madhav P. V. Kolluri* - "A 12-bit 500-ns Subranging ADC" - IEEE Journal of Solid-State Circuits, Vol. 24, No 6, Dezembro 1989, pgs. 1498 a 1506.
- *Makoto Imamura* - "A 1-Ms/s 16-bit Analog-to-Digital Converter" - IEEE Transactions on Instrumentations and Measurement, Vol. 39, No 1, Fevereiro 1990, pgs. 66 a 70.
- *Masao Hotta; Toshihiro Shimizu; Kenji Maio; Kazuo Nakazato e Shiichi Ueda* - "A 12 mW 6 bit Video Frequency A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. SC-22, No 6, Dezembro 1987, pgs. 939 a 943.
- *Michael J. Demler* - "High-Speed Analog-To-Digital Conversion" - 1ª ed., United Kingdom, Academic Press, Inc., 1991, 218 pgs.
- *Michael P. Flynn e Ben Sheahan* - "A 400-MSample/s, 6-b CMOS Folding and Interpolating ADC" - IEEE Journal of Solid State Circuits, Vol. 33, No. 12, Dezembro 1998, pgs. 1932-1938.
- *Michael P. Flynn e David J. Allstot* - "CMOS Folding A/D Converters with Current-Mode Interpolation"- IEEE Journal of Solid State Circuits, Vol. 31, No. 9, Setembro 1996, pgs. 1248-1257.
- *Michio Yotsuyanagi; Toshiyuki Etoh e Kazumi Hirata* - "A 10 bit 50 Mhz Pipelined CMOS A/D Converter with S/H" - IEEE Journal of Solid-State Circuits, Vol. 28, No 3, Março 1993, pgs. 292 a 300.
- *Paul R. Gray e Robert G. Meyer*, - "Analysis and Design of Analog Integrated Circuits", 3ª ed., USA, John Wiley & Sons, Inc., 1993, 792 pgs.
- *Pieter Vorenkamp e Raf Roovers* - "A 12-b, 60-MSample/s Cascaded Folding and Interpolating ADC" - IEEE Journal of Solid State Circuits, Vol. 32, No. 12, Dezembro 1997, pgs. 1876-1886.
- *R. Alini et al.* - "A 200 MSample/s Trellis-Coded PRML, Read/Write Channel with Analog Adaptive Equalizer and Digital Servo" - IEEE Journal of Solid State Circuits, Vol. 32, No. 11, Novembro 1997, pgs. 1824-1838.
- *Rob E. J. Van de Grift, Ivo W. J. M. Rutten e Martien Van der Veen* - "An 8 bit Video ADC Incorporating Folding and Interpolation Techniques" - IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 6, Dezembro 1987, pgs. 944 a 953.

- *Rob E. J. van de Grift e Rudy J. van de Plassche* - "A Monolithic 8-Bit Video A/D Converter" – IEEE Journal of Solid State Circuits, Vol. SC-19, No. 3, Junho 1984, pgs. 374-378.
- *Rudy J. van de Plassche* - "Integrated Analog to Digital and Digital to Analog Converters" - 1ª ed., Netherlands, Kluwer Academic Publishers, 1994, 501 pgs.
- *Rudy J. van de Plassche e Peter Baltus* - "An 8-bit 100-MHz Full-Nyquist Analog-to-Digital Converter" - IEEE Journal of Solid-State Circuits, Vol. 23, No. 6, Dezembro 1988, pgs. 1334 a 1344.
- *Rudy J. van de Plassche e Rob E. J. van de Grift* - "A High-Speed 7 Bit A/D Converter" - IEEE Journal of Solid State Circuits, Vol. SC-14, No. 6, Dezembro 1979, pgs. 938-943.
- *Sanroku Tsukamoto, William G. Schofield and Toshiaki Endo* - "A CMOS 6-b, 400-Msample/s ADC with Error Correction" - IEEE Journal of Solid-State Circuits, Vol. 33, No. 12, Dezembro 1998, pgs. 1939-1947.
- *Sidney Soclof* - "Design and Applications of Analog Integrated Circuits" - 1ª ed., USA, Prentice Hall, 1991, 820 pgs.
- *Simon J. Lovett, Marco Welten, Alan Mathewson e Barry Mason* - "Optimizing MOS Transistor Mismatch" - IEEE Journal of Solid-State Circuits, Vol. 33, No. 1, Janeiro 1998, pgs. 147-150.
- *Stephen Lewis e Paul Gray* - "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter" - IEEE Journal of Solid-State Circuits, Vol. SC-22, No 6, Dezembro 1987, pgs. 954 a 961.
- *Steven R. Nosworthy, G. Irving e H. Scott Fetterman* - "A 14-bit 80kHz Sigma-Delta A/D Converter: Modeling, Design, and Performance Evaluation" - IEEE Journal of Solid-State Circuits, Vol. 24, No 2, Abril 1989, pgs. 256 a 266.
- *Texas Instruments Incorporated* - "Understanding Data Converters" - 1ª ed., USA, Texas Instruments Incorporated, Julho 1995, 19 pgs.
- *Texas Instruments Incorporated* - "TLC5510, TLC5510A - 8-Bit High-Speed Analog-to-Digital Converters" - Texas Instruments Incorporated, September 1994 - Revised May 1999, 15 pgs. - Datasheet.
- *Udo Fiedler e Dieter Seitzer* - "A High Speed 8 Bit A/D Converter Based on a Gray-Code Multiple Folding Circuit" - IEEE Journal of Solid State Circuits, Vol. SC-14, No. 3, Junho 1979, pgs. 547-551.
- *V. Garuts, Y. Yu, E. Traa and T. Yamaguchi* - "A dual 4-bit 2-Gs/s full Nyquist analog-to-digital converter using a 70-ps silicon bipolar technology with borosinic-poly process and coupling-base implant" - IEEE Journal of Solid-State Circuits, Vol. 24, No. 2, Abril 1989, pgs. 216-222.
- *William T. Colleran e A. A. Abidi* - "A 10 b 75 Mhz Two Stage Pipelined Bipolar A/D Converter" - IEEE Journal of Solid-State Circuits, Vol. 28, No 12, Dezembro 1993, pgs. 1187 a 1199.
- *Y. Akazama et al.* - "A 400 Msps 8b flash AD conversion LSI" – ISSCC Dig. Tech. Papers, Vol. 30, 1987, pgs. 98-99.

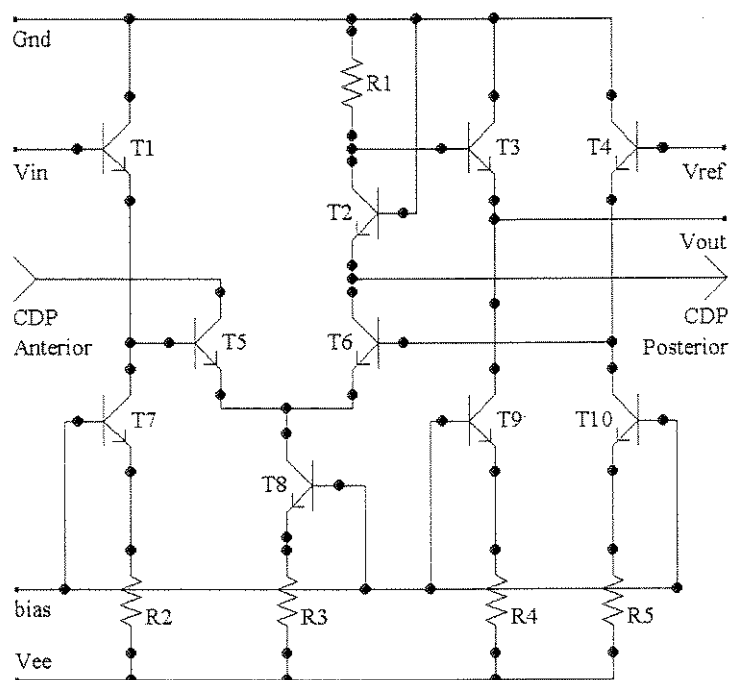
# ANEXO 1

# Amplificadores “Folding”

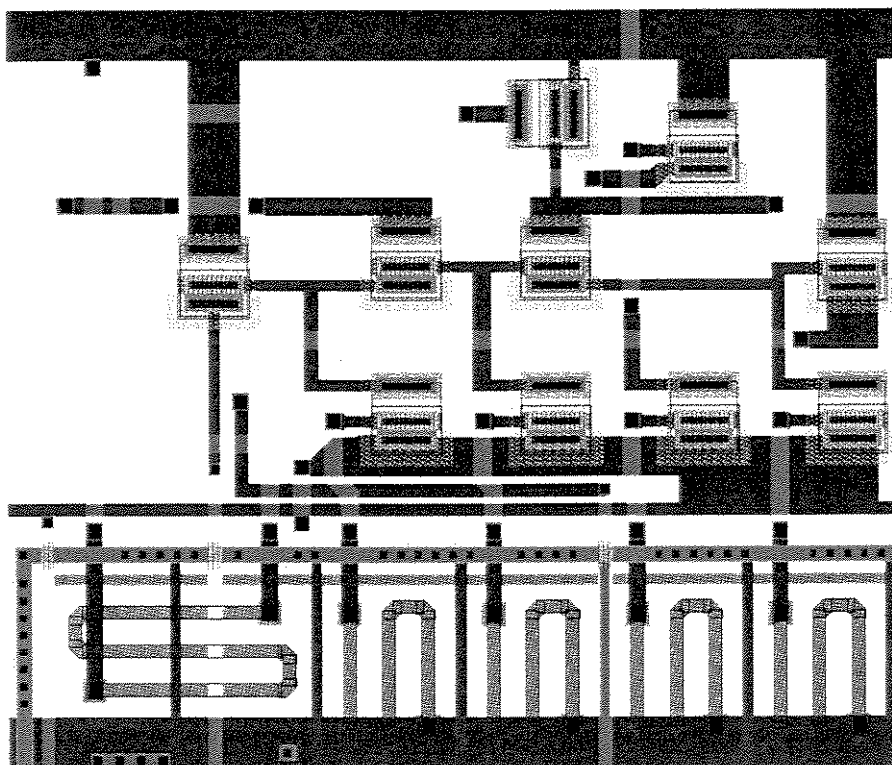


Obs: Existe uma estrutura “dummy” no amplificador “folding” 1.

# Amplificadores “Folding” - CDP

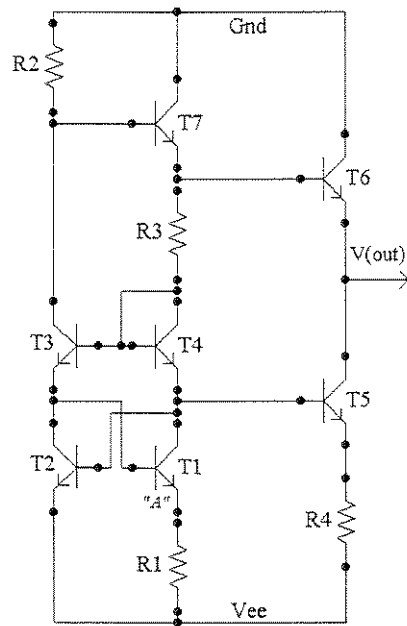


*Circuito*

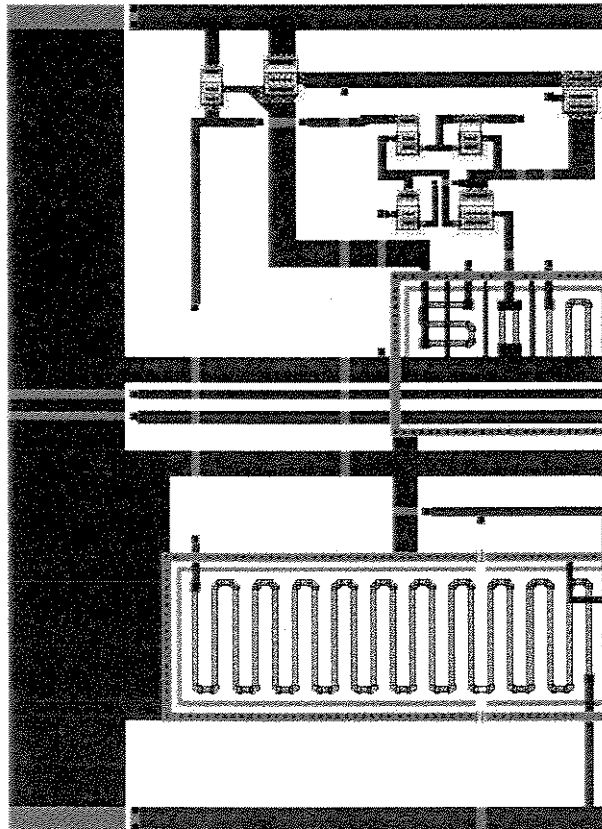


*“Layout”*

# Circuito da Fonte de Polarização

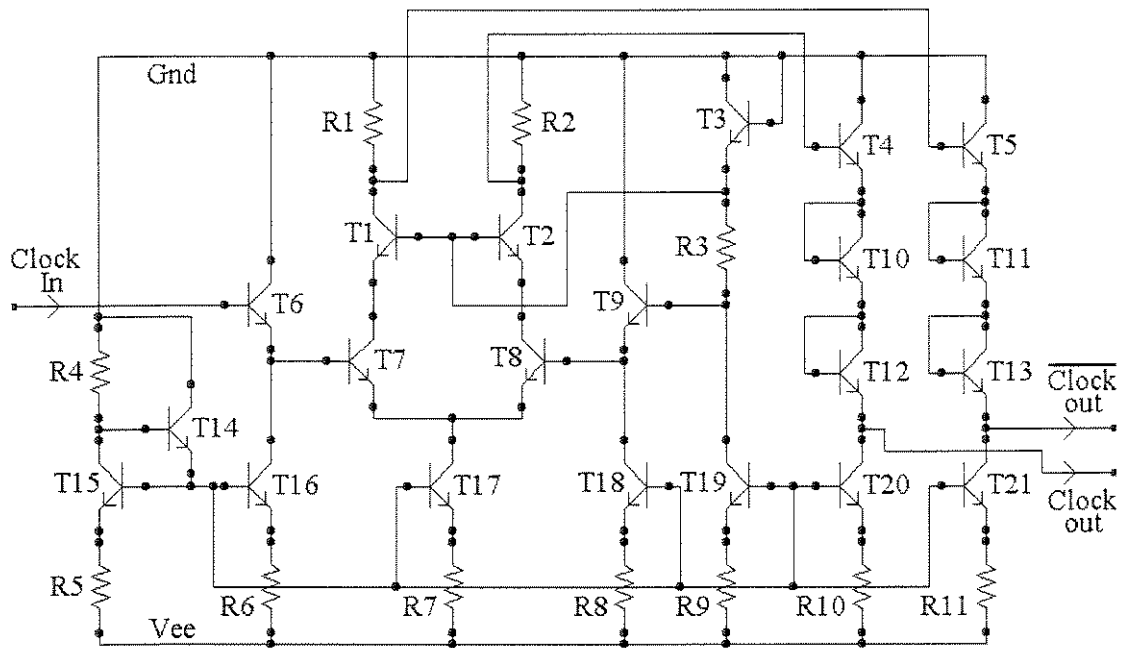


*Circuito*

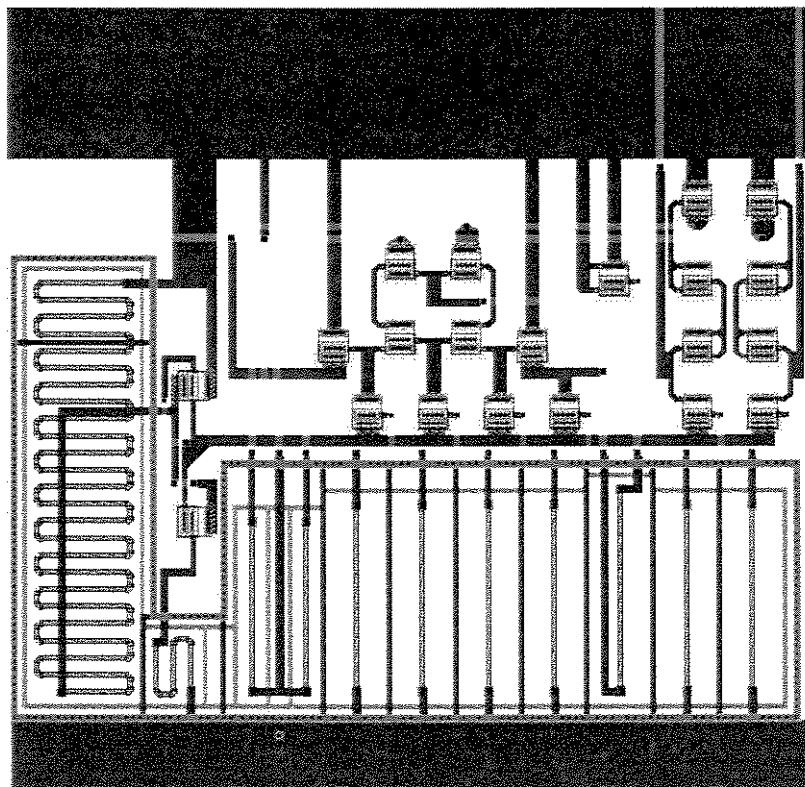


*"Layout"*

# Circuito de Entrada do Sinal de “Clock”

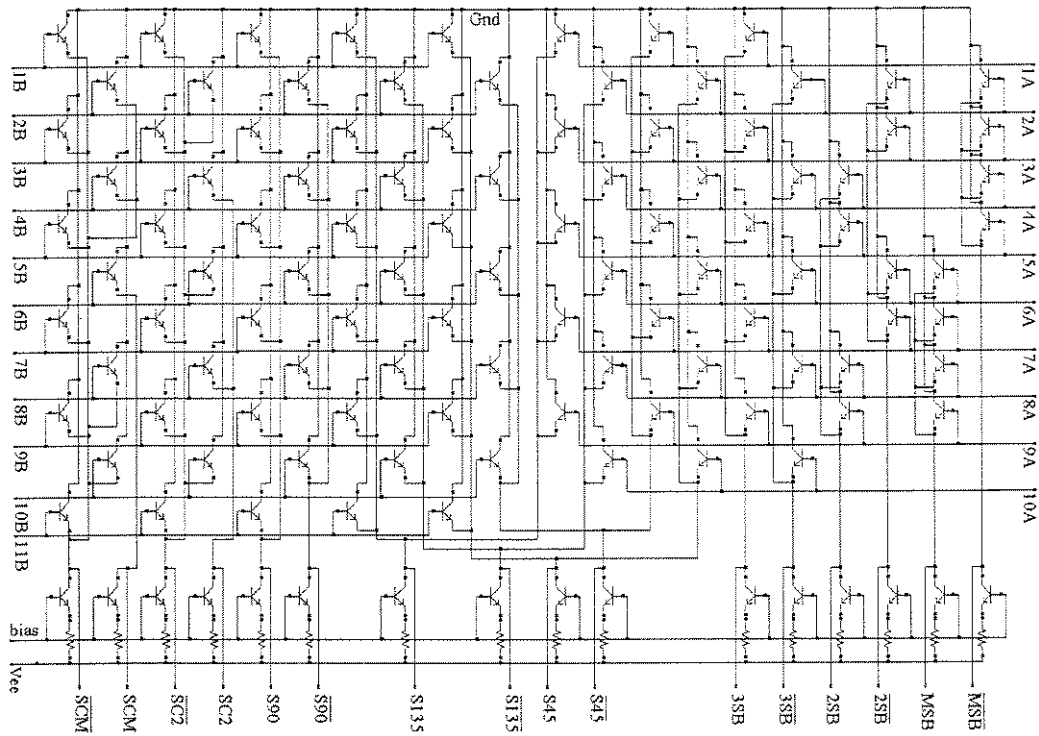


*Circuito*

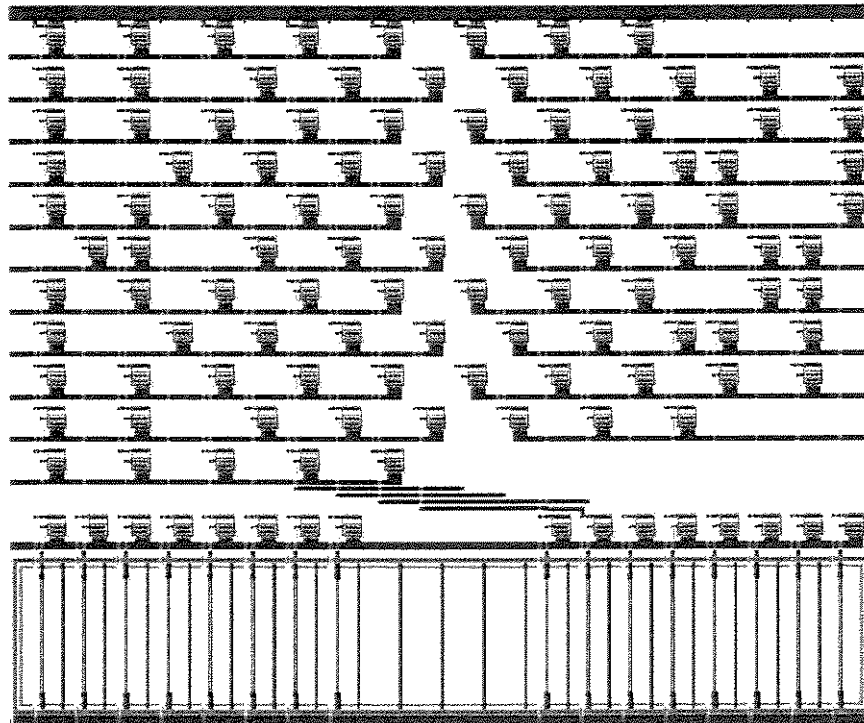


*“Layout”*

# Circuito do “Folding Encoder”



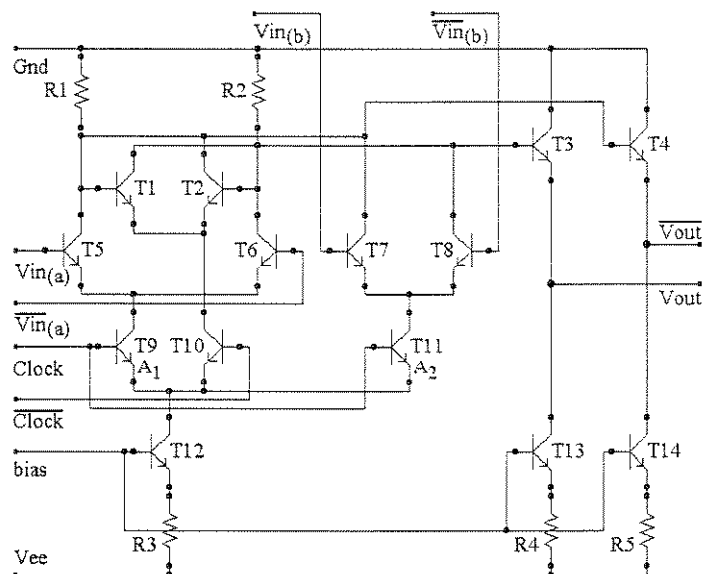
*Circuito*



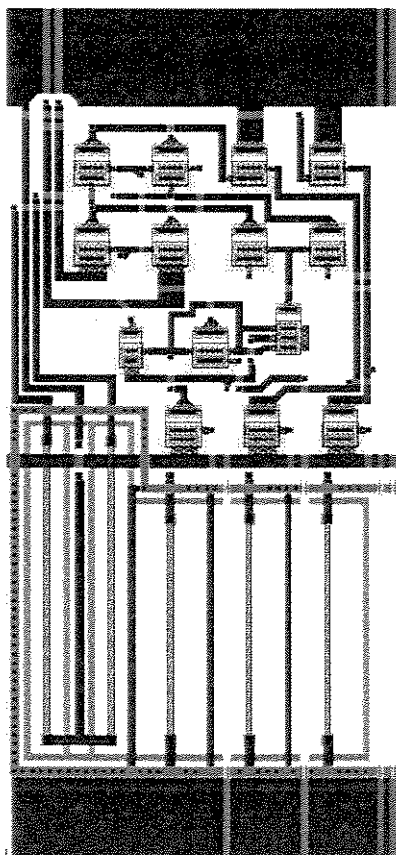
*“Layout”*



# Circuito do “Latch” Mestre Duplo de Interpolação

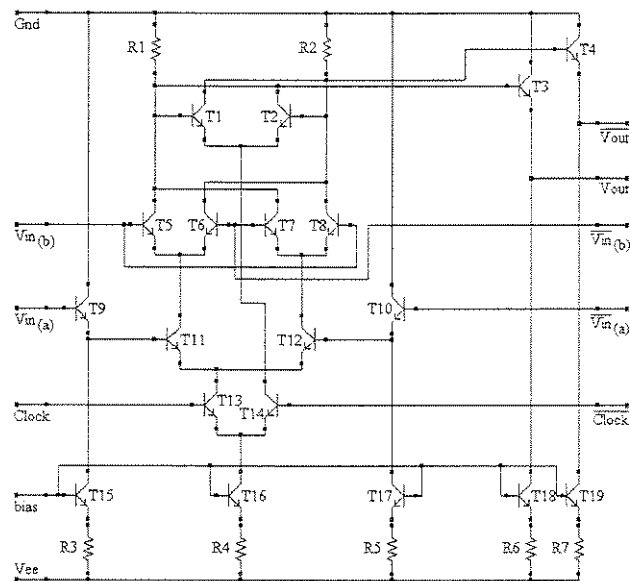


*Circuito*

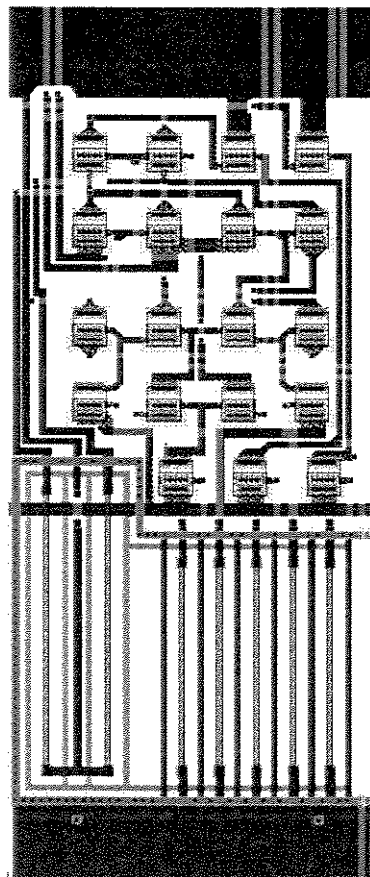


*“Layout”*

# Circuito da Porta “Ou-Exclusivo” com “Latch”

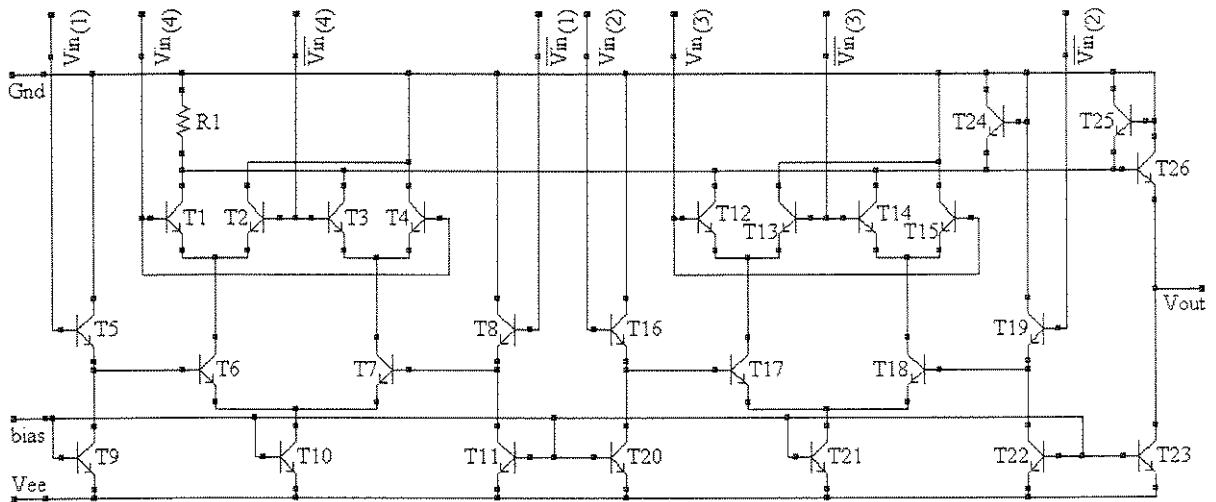


*Circuito*

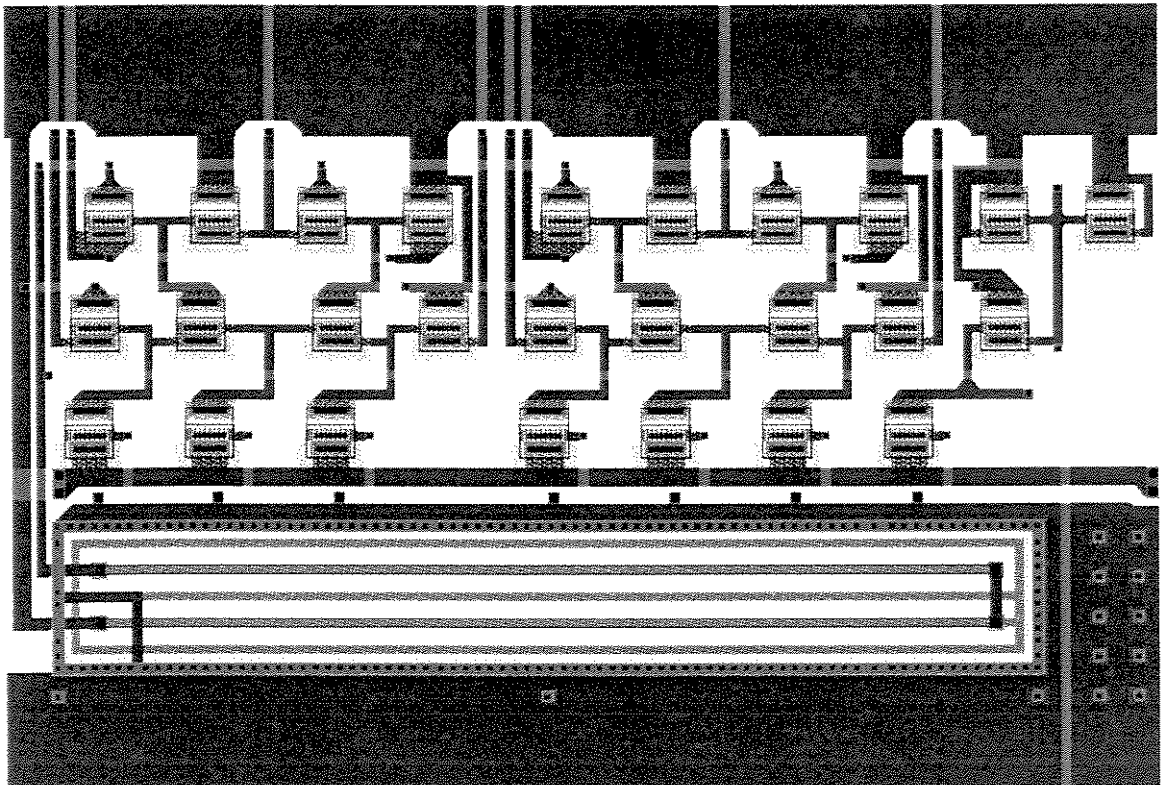


*“Layout”*

# Circuito de Correção de Erro

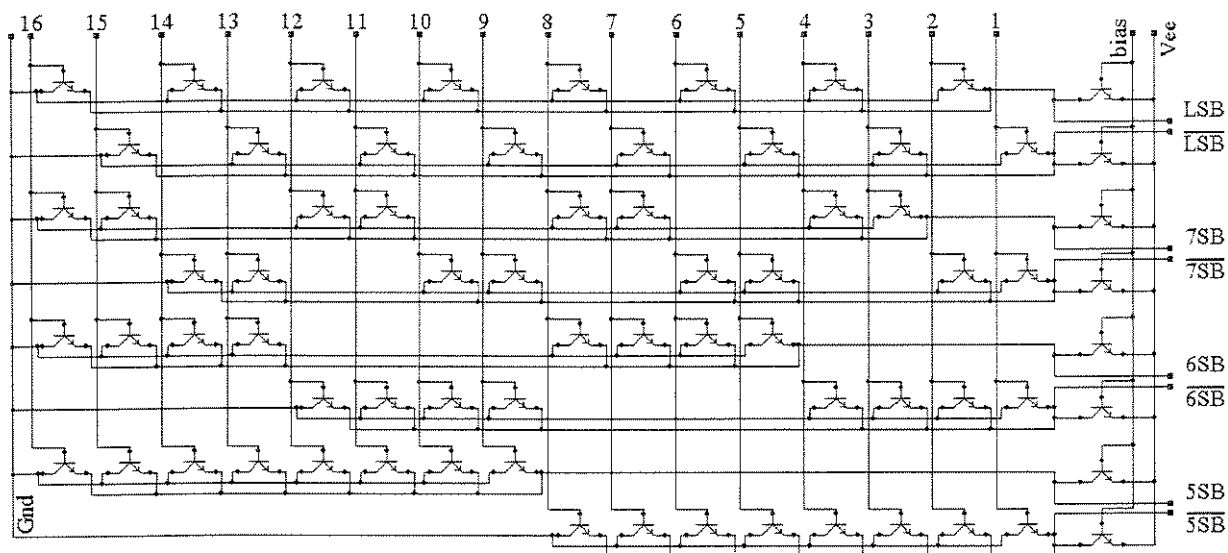


*Circuito*

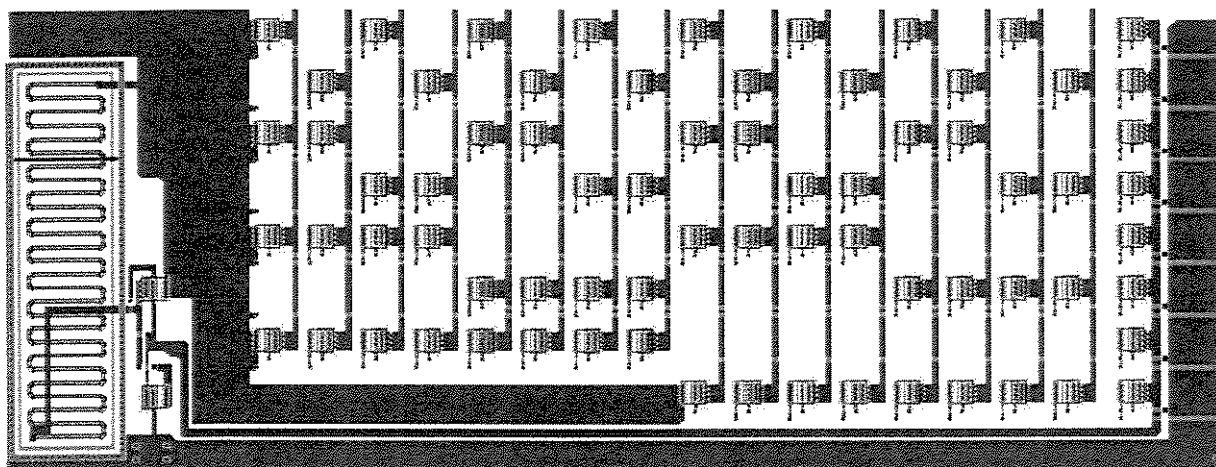


*“Layout”*

## Circuito Tipo ROM de Saída



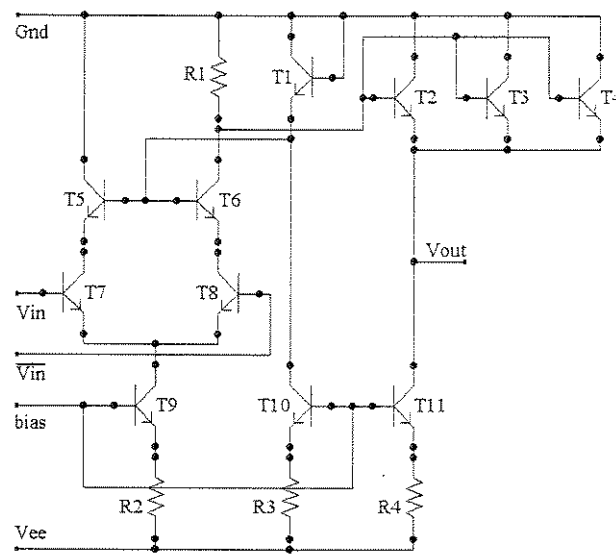
*Circuito*



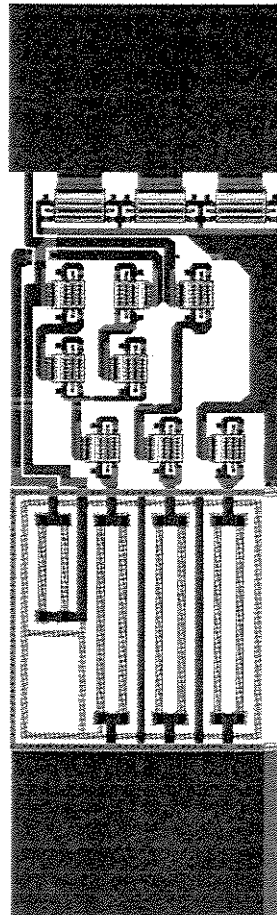
*“Layout”*

*Obs: O “layout” tem uma fonte de “bias” que não aparece no circuito*

## Circuito da Porta de Saída

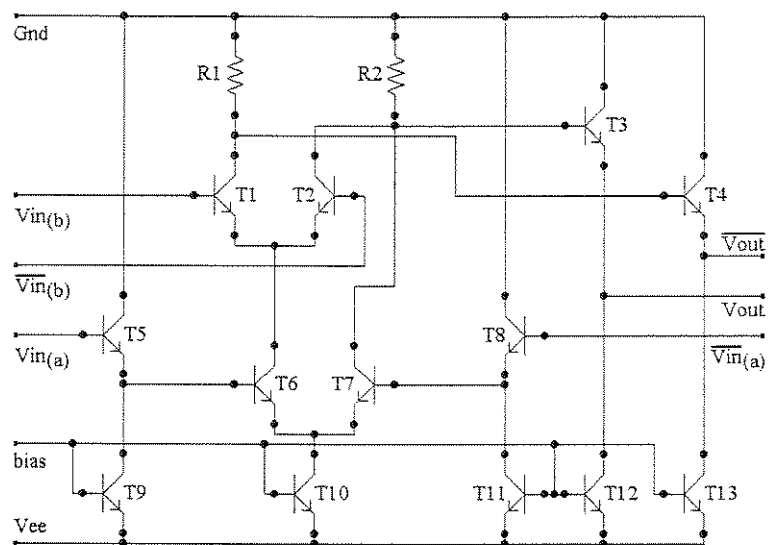


*Circuito*

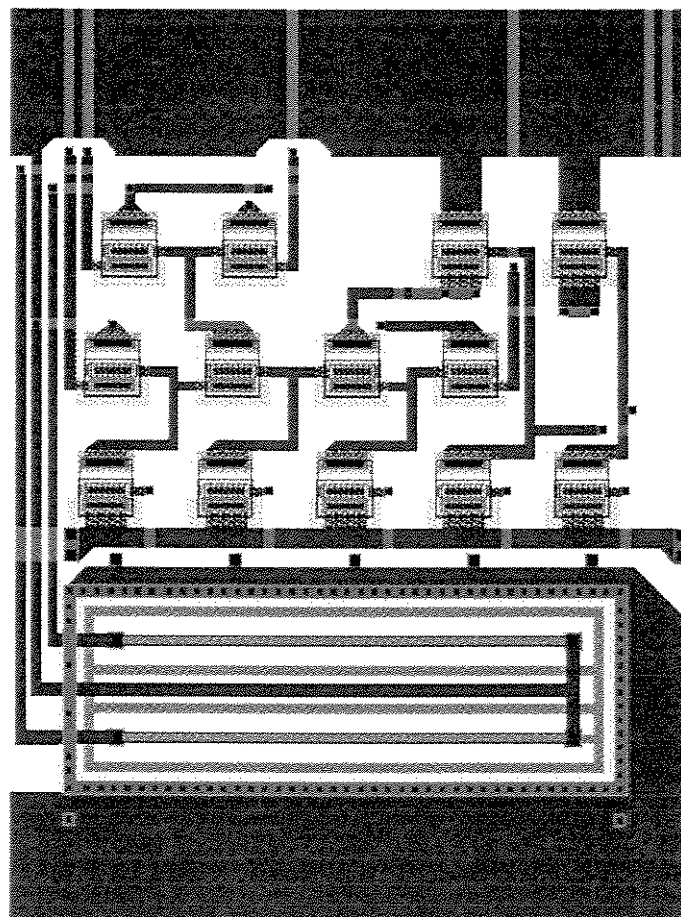


*“Layout”*

# Circuito da Porta AND com Duas Entradas

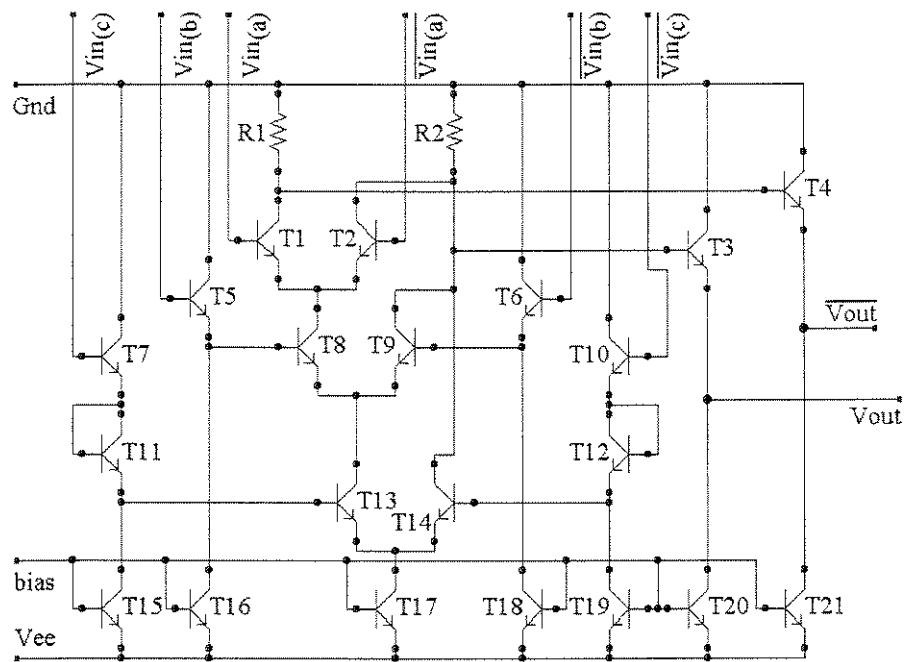


*Circuito*

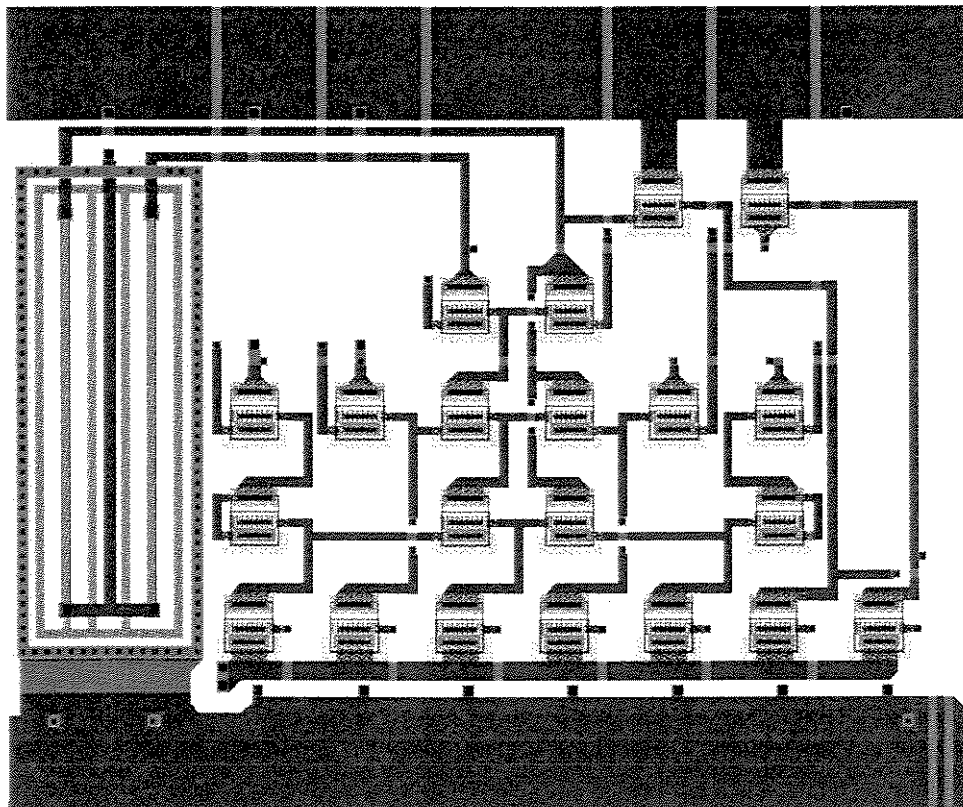


*“Layout”*

# Circuito da Porta AND com Três Entradas

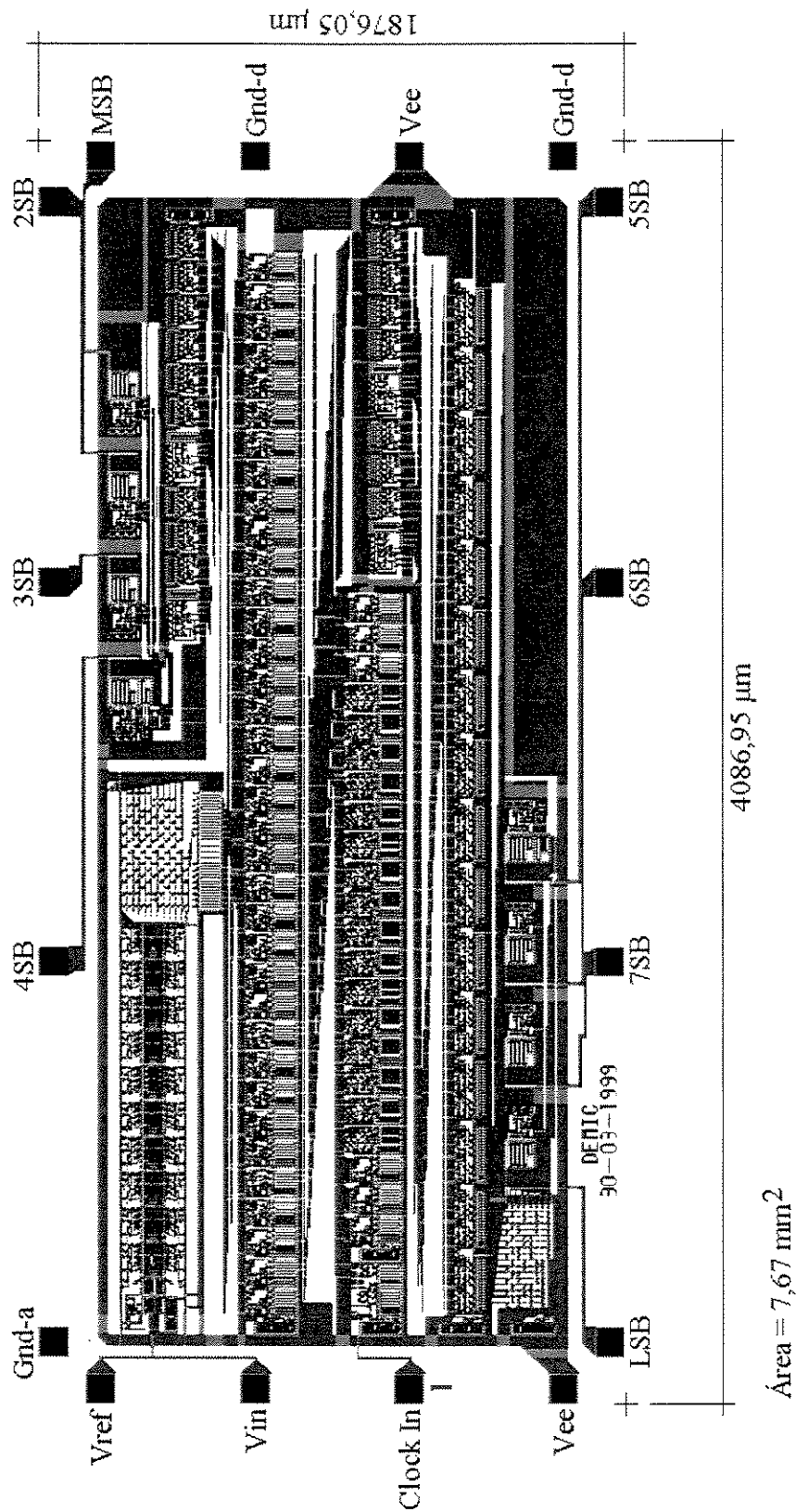


*Circuito*



*"Layout"*

# “Layout” Final do Conversor A/D de 8 Bits



## **ANEXO 2**

## Anexo 2

### Avaliação de Conversores A/D

#### *1 – Avaliação de Conversores A/D:*

##### *1.1 – Características Gerais:*

Um conversor A/D representa o sinal analógico de entrada em um número limitado de códigos digitais na saída. A Fig. 1 mostra um exemplo, onde cada código digital de saída representa uma fração do sinal analógico de entrada. É fácil perceber que o sinal analógico de entrada é contínuo, enquanto o código digital de saída é discreto. Portanto, existe um processo de quantização que introduz um erro (chamado de erro de quantização) [08, 17, 25 e 27]. Quando o número de códigos digitais de saída aumenta, a faixa (ou degrau) correspondente fica menor e a função de transferência se aproxima de uma linha reta ideal. Os degraus da curva de transferência são projetados para terem uma transição de tal maneira que o ponto médio de cada degrau corresponda a um ponto da linha reta ideal.

A largura de cada degrau é definida como um bit menos significativo (1 LSB) e define a resolução do conversor A/D porque é o valor que determina o tamanho das divisões possíveis do sinal analógico de entrada. Desta forma, a resolução do conversor A/D é expressa em número de bits do código digital de saída. Um conversor A/D com resolução de “n” bits, tem  $2^n$  possíveis códigos digitais de saída (que definem  $2^n$  degraus na curva de transferência).

Normalmente, é aplicado um “offset” de  $\frac{1}{2}$  LSB no sinal analógico de entrada, de maneira que o primeiro e o último degrau tenham somente metade da largura total dos demais degraus. Isto é feito para que a linha reta ideal (que reflete uma curva de transferência ideal) passe exatamente no ponto médio de cada degrau. Isto permite uma melhor representação do sinal

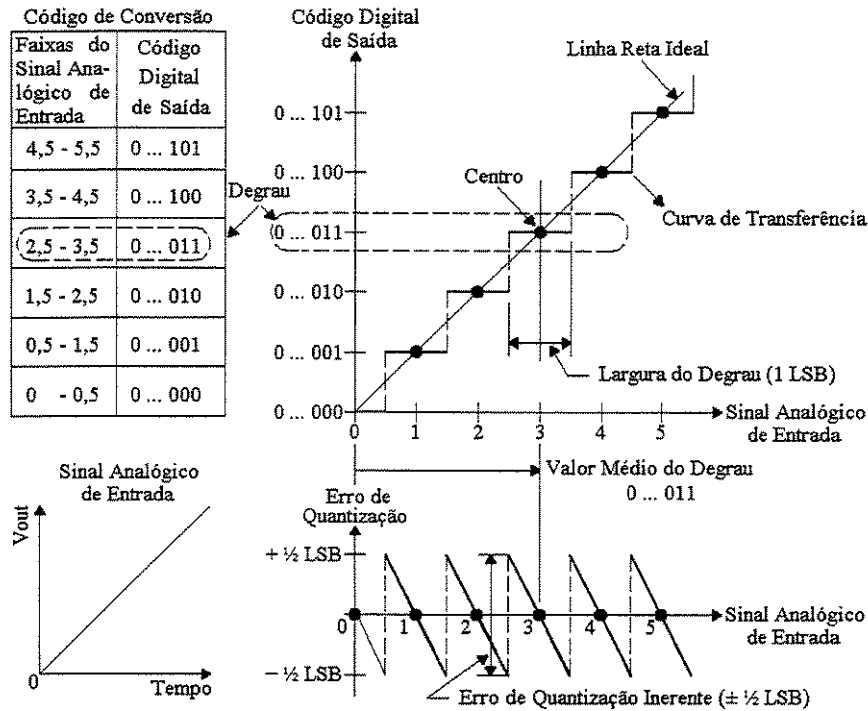


Figura 1 – *Curva de Transferência*

analógico de entrada pelo código digital de saída e, ainda, permite que o erro de quantização seja minimizado e fique na faixa ideal de  $\pm \frac{1}{2}$  LSB [17]. Portanto, uma vez que o primeiro degrau e o último degrau têm somente metade da largura total dos demais degraus, a faixa do fundo de escala (FSR – “full-scale range”) é dividida em  $(2^n - 1)$  degraus [25].

Resultando em:

$$1 \text{ LSB} = \text{FSR} / (2^n - 1) \quad (1)$$

## 1.2 – Quantização:

O sinal analógico de entrada de um conversor A/D é contínuo e o código digital de saída é discreto, com um número limitado de diferentes estados determinados pela resolução do conversor A/D. Isto significa que diferentes voltagens do sinal analógico de entrada serão representadas por um mesmo código digital de saída e parte da informação original será perdida. O sinal reconstruído na saída não será exatamente a reprodução do sinal de entrada. A diferença será o erro de quantização.

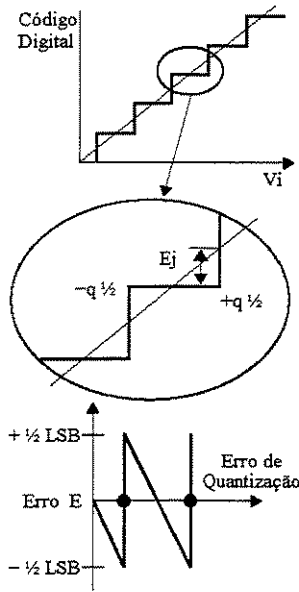


Figura 2 – Quantização

Definindo “q” como a largura de um LSB, a sua variação é de  $\pm \frac{1}{2}$  LSB ou seja,  $\pm \frac{1}{2} q$  (veja a Fig. 2). O erro em um degrau será:

$$E_j = (V_j - V_i) \quad (2)$$

O erro quadrático médio no degrau será:

$$\overline{E_j^2} = \frac{1}{q} \int_{-q/2}^{+q/2} E_j^2 \cdot dE = \frac{q^2}{12} \quad (3)$$

Sendo todos os degraus iguais, o erro quadrático médio será igual ao ruído quadrático médio [25]:

$$\overline{E_j^2} = \overline{N^2} = \frac{q^2}{12} \quad (4)$$

Considerando um sinal senoidal analógico de entrada igual a:

$$F(t) = A \cdot \text{sen}(\omega \cdot t) \quad (5)$$

O valor quadrático médio deste sinal será:

$$\overline{F^2} = \frac{1}{2\pi} \int_0^{2\pi} A^2 \cdot \text{sen}^2(\omega t) \cdot d\omega t = \frac{A^2}{2} \quad (6)$$

Como:

$$q = \frac{2 \cdot A}{2^n} = \frac{A}{2^{(n-1)}} \quad (7)$$

A razão sinal/ruído (“signal to noise ratio” – SNR) fica igual a:

$$\text{SNR} = 10 \cdot \text{Log}\left(\frac{\overline{F^2}}{\overline{N^2}}\right) = 10 \cdot \text{Log}\left(\frac{A^2/2}{A^2/3 \cdot 2^n}\right) = 6,02 \cdot n + 1,76 = (\text{dB}) \quad (8)$$

Este é o valor ideal da razão sinal/ruído (SNR) para um conversor A/D ideal de “n” bits.

Pela equação (8) percebe-se que cada bit adicional de resolução melhora a razão sinal/ruído (SNR) em aproximadamente 6 dB.

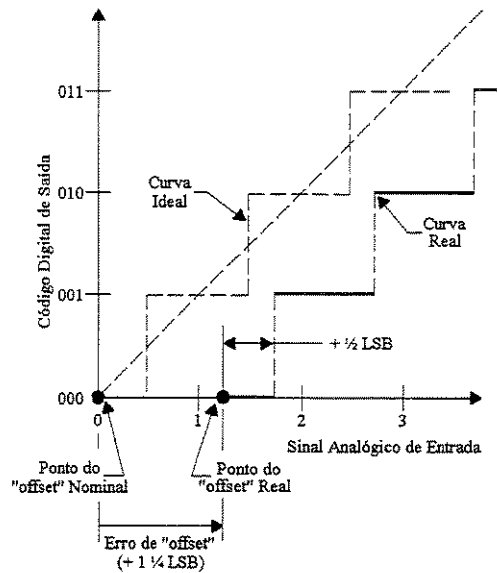


Figura 3 – Erro de “Offset”

### 1.3 – Análise Estática:

#### 1.3.1 – Características:

Os erros estáticos são os que afetam a precisão do conversor A/D quando é convertido um sinal estático (dc). Estes erros podem ser descritos por quatro termos básicos: erro de “offset”, erro de ganho, erro de não linearidade diferencial e erro de não linearidade integral.

#### 1.3.2 – Erro de “Offset”:

O erro de “offset” pode ser definido como a diferença entre o “offset” nominal e o “offset” real do conversor A/D (veja a Fig. 3). Este tipo de erro afeta todos os códigos digitais de saída da mesma forma e, também, é conhecido como “zero-scale error” [08 e 25].

#### 1.3.3 – Erro de Ganho:

O erro de ganho é definido como a diferença entre o ganho nominal e o ganho real na curva de transferência, após o erro de “offset” ter sido corrigido (veja a Fig. 4). No conversor A/D, o ponto de ganho é no meio do degrau da curva de transferência quando o sinal de saída digital está em fundo de escala. Este erro representa uma diferença entre a inclinação da curva de transferência real e da curva de transferência ideal e, como tal, corresponde a uma mesma

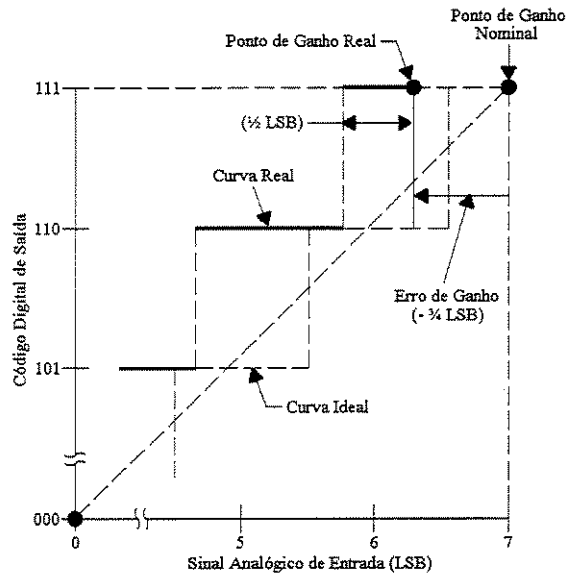


Figura 4 – Erro de Ganho

porcentagem de erro em cada degrau da curva de transferência [17 e 25].

#### 1.3.4 – Erro de Não Linearidade Diferencial (DNL):

O erro de não linearidade diferencial (também chamado apenas de linearidade diferencial) é a diferença entre a largura de um degrau real e o valor ideal de um 1 LSB. Portanto, se a largura (ou altura) de um degrau é exatamente 1 LSB, então o erro de não linearidade diferencial será igual a zero (veja a Fig. 5) [08, 17 e 25].

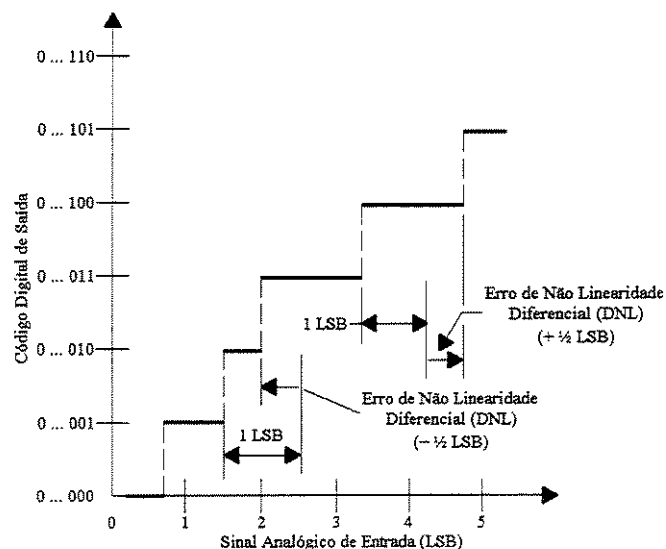


Figura 5 – Erro de Não Linearidade Diferencial (DNL)

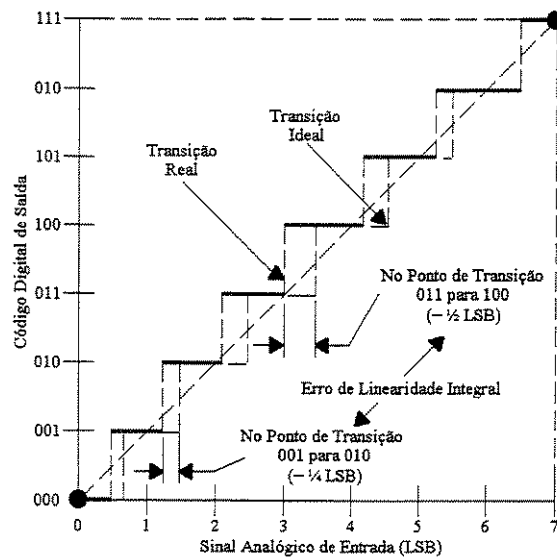


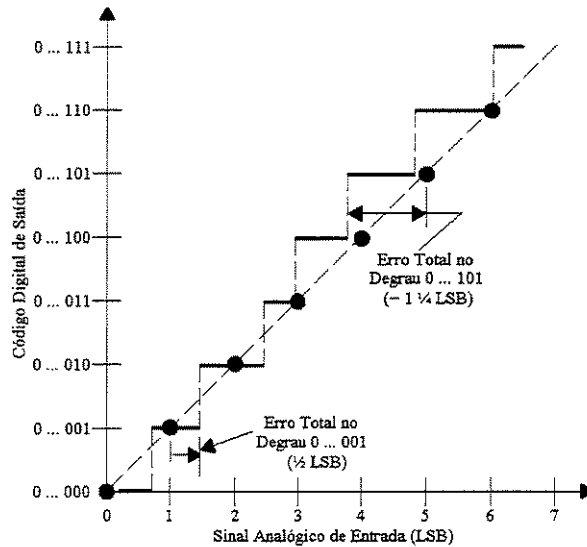
Figura 6 – *Erro de Não Linearidade Integral (INL)*

Se o DNL excede 1 LSB, existe a possibilidade do conversor A/D não ser monotônico. O conversor A/D é monotônico quando os códigos binários de saída incrementam (ou decrementam) e quando o sinal de entrada é continuamente incrementado (ou decrementado). Quando o DNL, no conversor A/D, excede 1 LSB, também existe a possibilidade de ocorrer perda de código, ou seja, um ou mais dos  $2^n$  códigos binários possíveis pode nunca aparecer na saída [25].

#### 1.3.5 – *Erro de Não Linearidade Integral (INL):*

O erro de não linearidade integral (também chamado apenas de linearidade integral) é o desvio dos valores da curva de transferência real em relação à linha reta (veja a Fig. 6) [08, 17 e 25]. Esta linha reta pode ser a que melhor se ajusta à curva de transferência real ou pode ser uma linha reta desenhada entre os pontos inicial e final da curva de transferência, desde que os erros de “offset” e de ganho tenham sido corrigidos. Este segundo método é o mais utilizado porque pode ser mais diretamente verificado e, também, é conhecido como “end-point linearity” [25].

Os desvios são medidos do ponto de transição de um degrau em relação à linha reta. Na curva de transferência ideal, o valor deve ser de  $\frac{1}{2}$  LSB, para a curva de transferência real; a

Figura 7 – *Erro Total*

diferença em relação a este valor é a medida do INL. De fato, o INL e o DNL são relacionados. O nome não linearidade integral origina-se da soma das não linearidades diferenciais do início até um degrau em particular, o que determina o valor da não linearidade integral naquele degrau [25].

#### 1.3.6 – *Erro Total:*

O erro total de um conversor A/D pode ser visto na Fig. 7 e corresponde ao máximo valor da diferença entre um valor analógico e o valor ideal do ponto médio do degrau. A medida inclui os erros de “offset”, de ganho, de não linearidade integral e, também, o erro de quantização.

### 1.4 – *Análise Dinâmica:*

#### 1.4.1 – *Características:*

Esta análise serve para avaliar a performance dinâmica do conversor A/D. A análise dinâmica pode ser realizada considerando-se alguns termos básicos: o número efetivo de bits, o teste da frequência de batimento e o erro de abertura.

#### 1.4.2 – *Número Efetivo de Bits:*

O número efetivo de bits representa a resolução de um conversor A/D ideal (livre de erros) com ruído de quantização igual aos erros totais do conversor sob teste [08 e 26]. O número

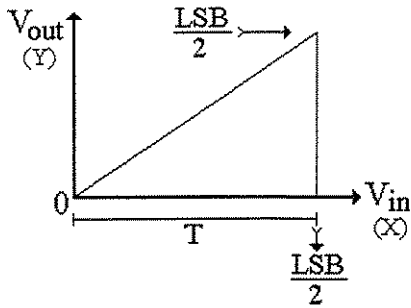


Figura 8 – Cálculo de  $V_{rms}$

efetivo de bits procura agregar a distorção e ruído do conversor em um único valor numérico. O número efetivo de bits é medido aplicando-se uma sinal senoidal ( $V_{in}$ ) na entrada do conversor A/D. Calcula-se, então, o sinal senoidal que melhor se ajusta ao sinal real de saída do conversor D/A (utilizado na reconstrução do sinal). A

seguir, é feita a diferença entre o sinal real reconstruído e o sinal senoidal calculado, deixando um sinal de erro que inclui o ruído de quantização e todas as outras imperfeições do conversor A/D. Deste sinal de erro, é calculado o valor quadrático médio que será o  $V_{rms}$ (erro-real). Esta diferença entre os dois sinais é equivalente a remover a frequência fundamental da transformada de Fourier do sinal reconstruído de saída do conversor D/A [26].

O cálculo do  $V_{rms}$  (erro-ideal) é feito aplicando-se uma rampa linear ( $V_{in}$ ) na entrada do conversor A/D. Como a curva de transferência de um conversor A/D ideal tem todos os códigos (degraus) exatamente iguais, o valor de  $V_{rms}$  (erro-ideal) é calculado para o primeiro código (degrau) da curva e o seu valor é extensivo para toda a curva. Na Fig. 8, a equação da reta é dada por:  $Y = a \cdot X + b$ . De fato, o  $V_{rms}$  (erro-ideal) é a medida do valor quadrático médio do erro de quantização. O valor de  $V_{rms}$  (erro-ideal) é calculado pela equação abaixo:

$$V_{rms}(\text{erro} - \text{ideal}) = \sqrt{\frac{1}{T} \int_0^T (a \cdot x)^2 dx} = \left[ \sqrt{\frac{a^2}{T} \frac{x^3}{3}} \right]_0^T = \frac{LSB}{2\sqrt{3}} = \frac{1}{2^{(n+1)}\sqrt{3}} = 0,001127637 \quad (9)$$

Sendo:  $a = 1$ ,  $b = 0$  e  $T = LSB/2 = 1/(2^n \cdot 2) = 1/2^{(n+1)}$  - sendo:  $n = 8$

E o valor do número efetivo de bits “n” é obtido pela equação abaixo [26], onde “n” é o número real de bits do conversor A/D.

$$n' = n - \log_2[V_{rms}(\text{erro-real}) / V_{rms}(\text{erro-ideal})] \quad (10)$$

Em [05], temos a introdução de uma figura de mérito extra, que seria a largura de banda

com resolução de n-bit, definida como a frequência do sinal de entrada ( $V_{in}$ ) na qual o número efetivo de bits tem um decréscimo de ( $n' - 0,5$ ). Até esta frequência, seria razoável considerar o conversor como tendo uma resolução de n-bits.

Para sinais de entrada ( $V_{in}$ ) com fundo de escala, temos a relação entre o número de bits efetivo ( $n'$ ) e a razão sinal ruído e distorção (SNDR), conforme [26] e [27]:

$$\text{O valor rms de uma onda senoidal é: } V_{rms}(\text{senóide-ideal}) = \frac{V_p}{\sqrt{2}} = \frac{V_{pp}}{2\sqrt{2}} \quad (11)$$

$$\text{No conversor de n-bits o valor é: } V_{rms}(\text{senóide-real}) = \frac{V_{pp}}{2\sqrt{2}} = \frac{2^n}{2} \frac{LSB}{\sqrt{2}} = \frac{2^{(n-1)}}{\sqrt{2}} \frac{LSB}{\sqrt{2}} \quad (12)$$

$$\text{O erro de quantização é: } V_{rms}(\text{erro de quantização}) = V_{rms}(\text{erro-ideal}) = \frac{LSB}{2\sqrt{3}} \quad (13)$$

A razão sinal/(ruído e distorção) é dada por:

$$SNDR = V_{rms}(\text{senóide-real})/V_{rms}(\text{erro-ideal}) = \frac{2^{(n-1)}}{\sqrt{2}} \frac{LSB}{\sqrt{2}} \frac{2\sqrt{3}}{LSB} = \frac{2^n \sqrt{3}}{\sqrt{2}} \quad (14)$$

$$\text{Expresso em dB, temos: } SNDR \text{ (dB)} = 20 \cdot [\log(\frac{\sqrt{3}}{\sqrt{2}}) + n \cdot \log 2] = 1,76 + 6,02 \cdot n \quad (15)$$

$$\text{Conforme [26], temos para fundo de escala: } SNDR \text{ (dB)} = 6,02 \cdot n' + 1,76 \quad (16)$$

#### 1.4.3 – Teste da Frequência de Batimento:

O teste da frequência de batimento (“beat frequency test”) [26] é realizado colocando-se o valor da frequência do sinal senoidal de entrada ( $f_{in}$ ) um pouco diferente da frequência de amostragem ( $f_s$ ) (veja a Fig. 9). Maior (ou menor), por exemplo, por uma diferença  $d_f$ , sendo  $d_f \ll f_{in}$ . Isto significa que o conversor A/D fará uma amostragem por período do sinal de entrada. Como a frequência do sinal de entrada é um pouco maior (ou menor) do que a frequência de amostragem, o instante amostrado irá avançar um pouco a cada amostragem. De fato, o sinal de entrada será efetivamente amostrado a uma razão igual a  $f_s/d_f$  [26]. O resultado é que o sinal de saída reconstruído é senoidal com a frequência igual a  $d_f$  (chamada de “beat frequency”). Este sinal senoidal, com as suas imperfeições, permite analisar qualitativamente o comportamento do

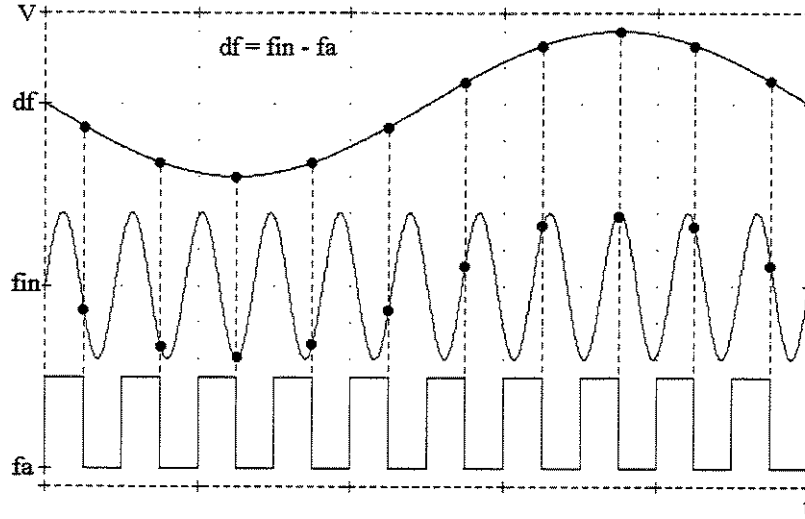


Figura 9 – Teste da Frequência de Batimento

conversor A/D.

#### 1.4.4 – Erro de Abertura:

O erro de abertura (“aperture error”) é causado pela incerteza no tempo em que a amostragem do sinal analógico de entrada vai do estado de amostragem (“sample”) para o estado de retenção (“hold”) (veja a Fig. 10). Esta variação é causada por ruído no sinal de “clock” ou no sinal analógico de entrada. O efeito do erro de abertura é causar mais uma limitação na máxima frequência do sinal analógico de entrada do conversor A/D porque definirá o máximo “slew rate” deste sinal [25]. Supondo um sinal analógico senoidal na entrada (V):

$$V = V_o \cdot \sin(2\pi \cdot f \cdot t) \quad (17)$$

O máximo “slew rate” ocorre no ponto de cruzamento por zero, sendo dado por:

$$\left. \frac{dV}{dt} \right|_{\max} = 2\pi \cdot f \cdot V_o \quad (18)$$

Para o erro de abertura não afetar a precisão do conversor A/D, ele deve ser menor do que  $\frac{1}{2}$  LSB no ponto de máximo “slew rate” [25]. Portanto, para um conversor A/D de “n” bits:

$$E_A = T_A \frac{dV}{dt} = \frac{1}{2} \text{LSB} = \frac{2 \cdot V_o}{2^{(n+1)}} \quad (19)$$

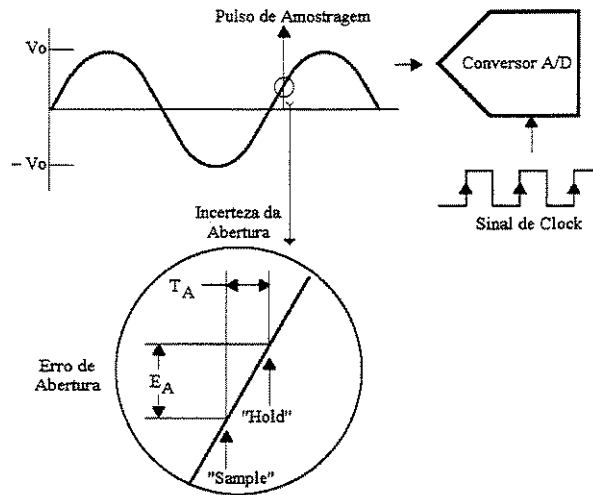


Figura 10 – *Erro de Abertura*

Substituindo alguns termos da equação (18) na equação (19), teremos:

$$\frac{2 \cdot V_o}{2^{(n+1)}} = 2 \cdot \pi \cdot f \cdot V_o \cdot T_A \quad (20)$$

O que resulta na máxima frequência limitada pelo erro de abertura:

$$f_{\max} = \frac{1}{T_A \cdot \pi \cdot 2^{(n+1)}} \quad (21)$$