

UNIDADE	BC
Nº CHAMADA	UNICAMP
	B47p
V	
TÍTULO	48601
	16-837/02
PREÇO	R\$ 11,00
DATA	
Nº CPD	

CM00165655-2

BIBID. 244043

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

B47p

Biazon Filho, Alcino José.

Projeto e construção de uma porta universal CMOS em lógica ternária / Alcino José Biazon Filho.-- Campinas, SP: [s.n.], 2001.

Orientador: Alberto Martins Jorge.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Circuitos integrados. 2. Lógica a múltiplos valores. 3. Circuitos eletrônicos - projetos. 4. Projeto lógico. 5. Transistores de efeito de campo de semicondutores de oxido metálico. I. Jorge, Alberto Martins. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

À minha família

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

207707 9-12

Agradecimentos.

Agradeço a Deus e à minha avó Antônia Pignata Sanches pelo apoio Espiritual.

- Ao meu orientador Alberto Martins Jorge, pela infinita paciência, incontestável didática e constante bom Humor.**
- Aos amigos, Telma, Paulo, Luciana, Caval, José Carlos, Jônatas, André, Ricardo, Ivo, Reinaldo, Ademilde e Fernando por agitar e despertar o que há de melhor em mim.**

Resumo

Neste trabalho desenvolvemos uma porta universal em lógica ternária através da álgebra de Post, utilizando-se dela pudemos desenvolver alguns circuitos conhecidos da lógica binária como Flip-Flops e Somadores. Esses circuitos foram simulados em SPICE e seu Lay-Out desenvolvido utilizando-se ferramentas como Tanner e LVS, para a construção de um circuito Integrado utilizamos uma Foundry que já conhecíamos e que possuía uma grande confiabilidade que foi a AMS CYE em 0.8 μm . Para os testes dos circuitos construídos utilizamos as instalações do Laboratório de Medidas (DEMIC/UNICAMP) com seus equipamentos ligados via GPIB e desenvolvemos instrumentos virtuais (V.I.) via Labview que pudessem controlar esses equipamentos e gerar alguns sinais necessários para a obtenção destas medidas. Comprovamos durante os testes a viabilidade das portas Topo (deslocador ternário), Alfatopo (mínimo entre duas variáveis ternárias, deslocada de um nível lógico) e do flip-flop (com o funcionamento idêntico ao tradicional tipo D) ternário.

Abstract

In this work we developed a universal gate in ternary logic through Post algebra; using this gate we could develop some well known circuits from binary logic like Flip-Flops and Adders. These circuits were simulated using Spice and the Lay-Out was developed using tools like Tanner and LVS; to construct the integrated circuit we use a foundry that we already knew as reliable, that was the AMS CYE, in 0.8 μm . To test the circuits we used the facilities of the Measurement Laboratory (DEMIC/UNICAMP) and the equipment's were linked via GPIB; we developed virtual instrumentation (V.I.) using Labview to control these equipment's and generate some necessary signals to obtain the final results. We proved during this tests the viability of the gates Topo (Ternary shifter), Alfatopo (minimum among two ternary variables, shifted in one logic level) and Flip-flop (identical of traditional type D) ternary.

Índice

Capítulo 1	1
1 - Introdução Teórica	1
1.1 - Introdução	1
1.2 - Álgebra de Post	2
1.3 - Circuito Alfatopo	6
1.4 - Flip-Flop	8
1.5 - Somador Simples de 3 Valores	14
1.5.1 - Processador Ternário	14
1.5.2 - Síntese de $S(A,B)$	14
1.5.3 - Síntese da Função V de Transporte	16
1.5.4 - Síntese de funções através do Método dos Operadores	19
Capítulo 2	23
2 - Circuitos Simulados/Resultados Esperados	23
2.1 - Alfatopo	24
2.2 - Flip-Flop	27
2.3 - Flip-Flop funcionando como Togle Switch	29
2.4 - Quatro Flip-Flops colocados como Shift Register Ternário	30
2.5 - Somador	31
Capítulo 3	33
3.1 - Circuito Integrado Fapesp73.Dem (Lay-Out e Sistemas de Medidas)	33
3.2 - Montagem Experimental	35
3.3 - GPIB	37
3.3.1 - Tipos de dispositivos GPIB	38

3.3.2 - Linhas e sinais GPIB	39
3.4 - Características físicas e elétricas	42
3.5 - Requisitos de configuração	43
3.6 - IEEE 488.2 e SCPI	44
3.7 - LAB - PC 1200	45
3.8 - Instrumentos Virtuais Utilizados	45
Capítulo 4	49
4 - Resultados Obtidos	49
4.1 - Topo	49
4.2 - Flip-Flop	51
4.3 - Toggle Switch	55
4.4 - Somador	57
Conclusão	59
Bibliografia	61
Anexo A	63
Anexo B	83

Capítulo 1

1 Introdução Teórica.

1.1 Introdução.

O atual desenvolvimento dos computadores está intimamente ligado a uma poderosa ferramenta digital binária que pode ser considerada como a alma do funcionamento dessas máquinas, a álgebra de Boole [Tocci98, Taub82]; atualmente, devido ao aumento da complexidade dos circuitos, exige-se algo mais poderoso e ao mesmo tempo intrinsecamente mais complexo para suprir suas limitações.

Quando dizemos que os circuitos estão aumentando em complexidade devemos perceber que o grande problema é o aumento das interconexões.

A proposta principal deste trabalho é o desenvolvimento de circuitos digitais em multi-valores com ênfase para os circuitos em três valores; buscamos mostrar a viabilidade destes circuitos como elementos simplificadores dos circuitos atuais.

Devido a crescente complexidade dos circuitos verificamos também um aumento da complexidade das conexões internas dos “chips” e entre os “chips”, como podemos observar nos modernos encapsulamentos dos “chips” comerciais; estes se mostram cada vez mais complexos devido a esse aumento do número de interconexões e conseqüente consumo de área do chip relativa aos “Pads” (área de “chip” reservada para a soldagem do silício com os contatos do encapsulamento responsáveis pela introdução ou retirada dos sinais no chip).

Pode-se demonstrar que em três ou mais valores estaremos processando uma quantidade maior de informações que com a lógica binária utilizando a mesma quantidade de contatos do que a utilizada em dois valores.

Em se tratando de um desenvolvimento de circuitos digitais nós não poderíamos ficar indiferentes a grande potencialidade dos transistores CMOS nas aplicações digitais; esta é a razão de escolhermos os elementos CMOS para a confecção dos circuitos desta tese.

Em relação a potencialidade dos elementos CMOS destacamos seu baixo consumo de potência e grande capacidade de chaveamento.

1.2 Álgebra de Post

Há a necessidade de explanarmos as bases teóricas da álgebra de Post para que fique claro a sua utilização mais adiante nessa dissertação [Epstein74, Serran96, Lukasiewicz20, Post20, Serran97].

A Álgebra de Post se aplica a um conjunto ordenado de objetos distintos $\{x_i\}$, em quantidade finita ($n+1$ elementos, sendo n um número natural), no qual se definem duas operações: uma delas unária, a negação, e a outra binária, a disjunção ou chamada de maximização. São definidas duas operações:

1- *Rotação cíclica unária ou complementação*: também conhecida como negação cíclica de Post, que por comodidade de representação indicaremos por uma barra superior adicionada ao símbolo do elemento que é complementado como segue abaixo.

$$\overline{t_i} = \begin{cases} t_{i+1} & \text{se } i \neq n \\ t_0 & \text{se } i = n \end{cases} \quad (\text{I.1})$$

Como ilustração observemos o caso de $n=2$ (três objetos, ou valores, t_0 , t_1 , e t_2):

$$\overline{t_0} = t_1 \qquad \overline{t_1} = t_2 \qquad \overline{t_2} = t_0 \quad (\text{I.2})$$

Para o caso de $n=1$ observa-se a complementação típica da álgebra de Boole.

A negação cíclica de Post se dá no sentido horário. Em muitas expressões ocorrem complementações duplas e até em quantidade maior sobre o mesmo elemento. Para simplificar a notação, adotamos o recurso de indicar o deslocamento no sentido anti-horário por uma barra inferior ($\underline{t_i}$).

2- *Disjunção binária ou Maximização* (\vee), isto é realizado entre dois valores do conjunto P_n ,

$$t_i \vee t_j = \text{máx}(t_i, t_j) \quad (\text{I.3})$$

ou seja trata-se de um “ou” entre t_i e t_j . Podemos também definir a conjunção “e” (\wedge), como é definido pela lógica clássica:

$$t_i \wedge t_j = \text{mín}(t_i, t_j) \quad (\text{I.4})$$

Através do mínimo podemos definir os operadores OP_k :

$$OP_0 = \overline{OP_1} = \overline{\overline{OP_2}} = \dots = \overline{OP_{n-2}} = \overline{OP_{n-1}} = \wedge = \min(t_i, t_j) \quad (I.5)$$

Esta dissertação foi desenvolvida para a lógica em três valores [Thoidis98] como maneira de investigação do seu comportamento com circuitos reais; então a partir de dois conjuntos A e B ordenados de objetos que possuam os mesmos elementos:

$$A = \{a, b, c\} \quad (I.6)$$

$$B = \{a, b, c\} \quad (I.7)$$

onde $c > b > a$

O operador mínimo (\wedge , “e”), OP_0 a partir desse momento será chamado de α , OP_1 de operador β e OP_2 de operador γ . Podemos mostrar as tabelas da verdade dos operadores α , β e γ .

Operador Alfa				Operador Beta				Operador Gama								
				A				A								
				A				A								
				A				A								
α		a	b	c	β		a	b	c	γ		a	b	c		
a		a	a	a	a		a	b	c	a		a	a	c		
B		b	a	b	b	B		b	b	b	B		b	a	b	c
		c	a	b	c			c	b	c			c	c	c	c

Tabela I.1

Tabela I.2

Tabela I.3

Para maiores detalhes da teoria básica e das propriedades da álgebra consultar a bibliografia [Serran96].

Para exemplificar e mostrar que as funções comuns podem (e sempre poderão) ser colocadas em função de Alfatopos e Topos utilizaremos a função A_0 ; esta função é constituinte do bloco Somador que será mostrado mais adiante neste capítulo (pág. 15).

A equação determinada através de inferência foi:

$$A_0 = B\beta(A\alpha 1)$$

Quando colocamos uma função Topo e Base na mesma equação não ocorre nenhuma alteração no resultado final.

$$A_0 = \overline{B\beta(A\alpha 1)}$$

Através dos operadores OPk (Teorema de Morgan para multi-valores) [Serran96], transformamos o beta em alfa, e assim o Alfatopo fora do parênteses já está definido.

$$A_0 = \overline{\underline{B\alpha(A\alpha 1)}}$$

Para completarmos temos que rearranjar o alfa interno do parênteses para que este torne-se Alfatopo; sabemos que um elemento Base é equivalente à um duplo Topo.

$$A_0 = \overline{\underline{\underline{B\alpha(A\alpha 1)}}}$$

Agora podemos desenhar o circuito usando somente elementos Alfatopo e Topo; este circuito é a figura 1.4 deste capítulo. No caso de outras funções o procedimento é exatamente o mesmo, podemos colocar Topos e Bases como foi efetuado nesta equação em elementos que estão entre parênteses e colchetes no meio da equação; isto não altera o resultado final, chegando à equação desejada.. [Biazon00, Martins99]

1.3 Circuito Alfatopo.

Abaixo está representado o circuito Alfatopo desenvolvido para esta dissertação; esta é a versão final que foi simulada [Spice95, Serran96, Allen93 Cap. 2, pp. 21-23]. Os resultados dessas simulações encontram-se no capítulo 2.

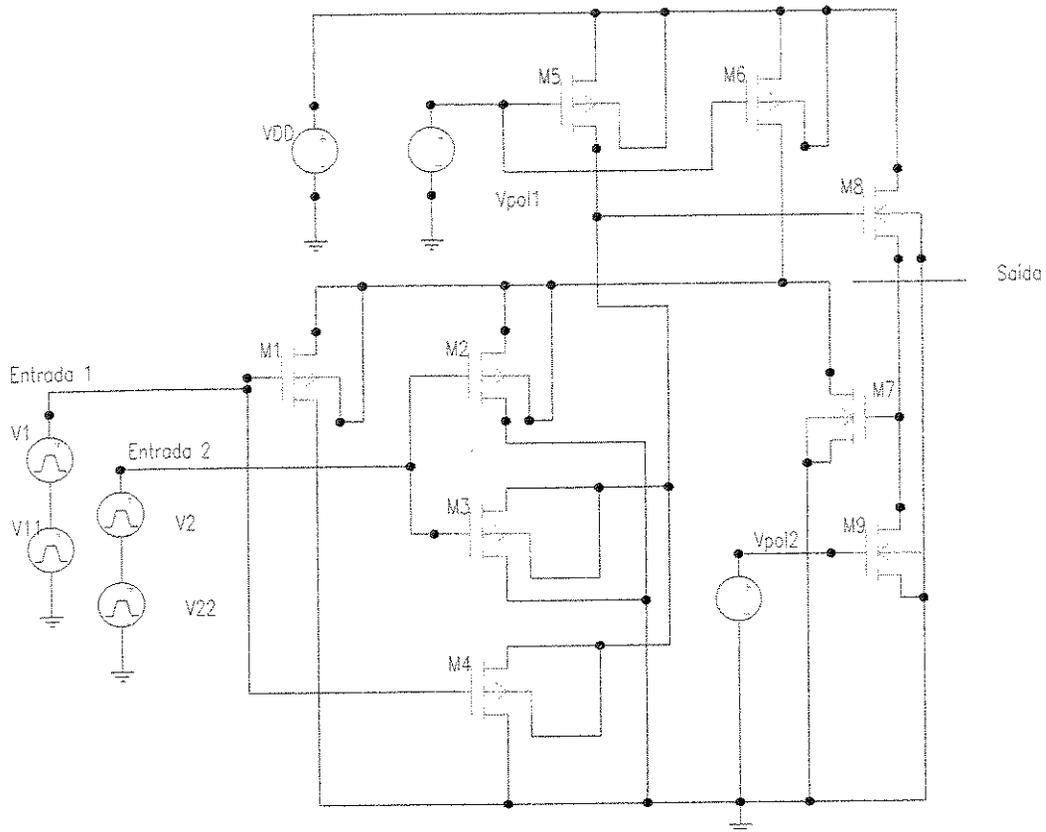


Figura 1.1

Cabe aqui uma pequena explicação das estruturas que podem ser observadas no circuito, as fontes V1 e V11 da entrada 1 são utilizadas para que possamos gerar o sinal em três valores no nosso simulador, assim como as fontes conectadas na entrada 2. Esses sinais em três valores são aplicados nas portas dos transistores M1 e M2; esses sinais são também aplicados nos transistores M3 e M4 que tem a função de deslocadores de nível no nosso

circuito. Os transistores M5, M6 e M9 tem a função de fontes de corrente contínua que podemos ajustar externamente através das fontes de tensão Vpol1 e Vpol2.

Essas tensões de polarização foram intencionalmente introduzidas para que pudéssemos ajustar no CI as características de polarização; assim poderíamos evitar alguma dispersão do processo e teríamos alguma liberdade no circuito para verificarmos se seria possível melhorar o sinal de saída desse circuito. Vale a pena destacar nesse ponto que este circuito e os demais deste trabalho funcionam em um nível de tensão muito baixo, caracterizando um baixo consumo de potência levando-se em conta que os circuitos são construídos utilizando-se elementos CMOS.

Como o nome já está sugerindo, este circuito executa a operação alfa cuja tabela da verdade já foi mostrada anteriormente, fornecendo também um deslocamento Topo nesse sinal. O Alfatopo é um elemento universal dentro da lógica multi-valores porque através dele poderemos construir qualquer uma das portas da lógica, funcionando exatamente como as portas NAND e NOR para a lógica binária.

Observamos durante o desenvolvimento teórico que teríamos a necessidade da criação de duas outras portas para que possamos completar a lógica em três valores; essas portas deveriam funcionar como um Deslocador Topo (Negação cíclica de Post) e um Deslocador Base (Negação cíclica anti-horária). Durante a concepção do circuito pudemos observar que o Topo (para simplificação nos referiremos assim ao Deslocador Topo daqui para frente) poderia ser desenvolvido apenas utilizando a porta Alfatopo com a exclusão da segunda entrada (ou seja excluindo-se os transistores M2 e M3)

A criação do Deslocador Base mostrou-se inviável porque haveria a necessidade da utilização de transistores de depleção que não estão disponíveis na tecnologia que seria

utilizada (CMOS 0.8um CYE); optamos então pela dupla negação para a obtenção do operador Base.

Durante o desenvolvimento dos circuitos mais complexos desta dissertação sentimos a necessidade do desenvolvimento de novas portas Alfatopo, mas que permitissem uma terceira entrada para efetuarmos alguns testes no circuito do Flip-Flop e a efetiva obtenção do nosso Somador ternário. Essa porta foi chamada de Alfa3topo e possui as mesmas características das portas Topo e Alfatopo com apenas a inclusão de mais uma entrada que usará mais dois transistores de canal P.

1.4 Flip-Flop

Para o desenvolvimento do Flip-Flop tipo D ternário foi utilizado como inspiração inicial o Flip-Flop tipo D binário com portas NAND que está mostrado na figura abaixo [Taub82], [Tocci98]:

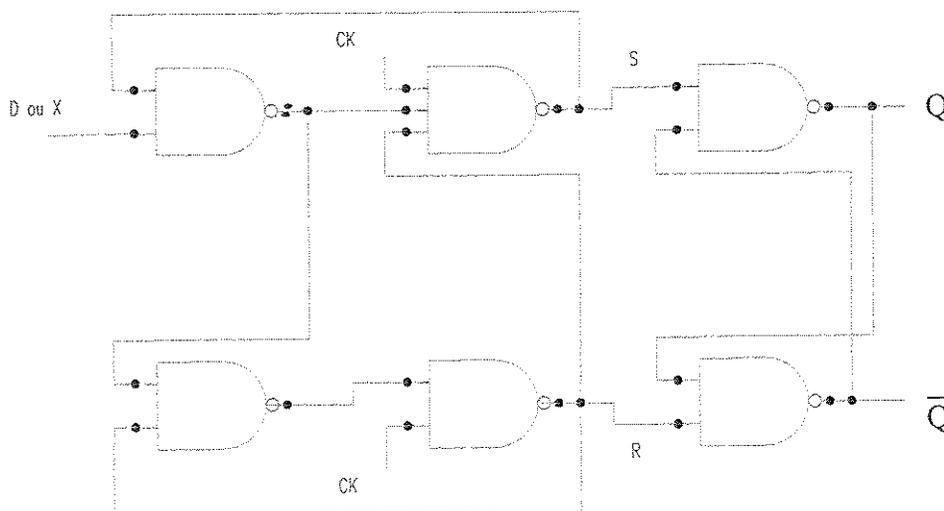


Figura 1.2

Para a obtenção do Flip-Flop em multivalores acrescentamos a terceira fileira de elementos Alfatopo e procuramos uma configuração que nos desse o resultado desejado na saída do sinal.

O Flip-Flop desenvolvido é do tipo D sensível a borda de subida e está detalhado na figura 1.3:

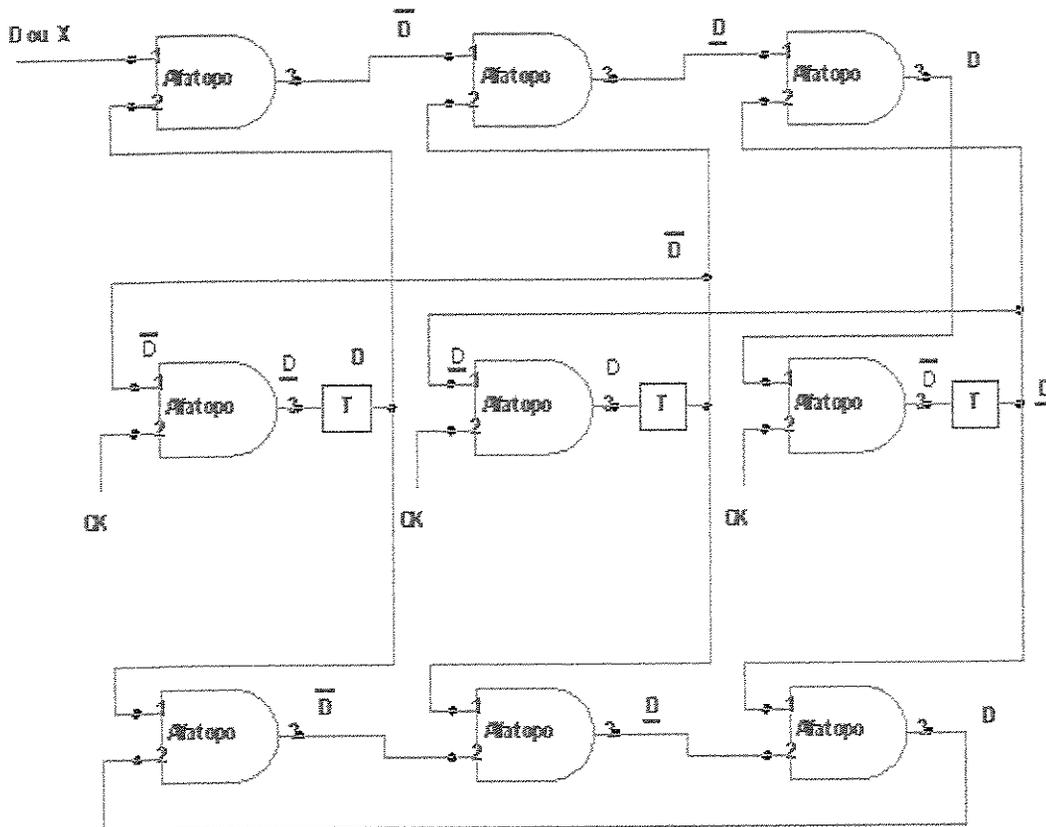


Figura 1.3

Na figura D representa o dado no momento inicial da análise do circuito e X representa um valor aleatório deste dado depois de estabilizado o “clock”; fazemos uso deste recurso para chegarmos à equação que comprova que este Flip-Flop será carregado com o sinal desejado somente quando este estiver na borda positiva do “clock” e que no

restante do tempo este dado permanecerá inalterado independente do dado introduzido no circuito.

Chamamos nesta dissertação os três Alfatopos superiores de “fila superior” e assim sucessivamente de “fila intermediária” e “fila inferior” ou Flip-Flop principal.

Podemos observar neste circuito que existem apenas duas entradas: uma delas representa o dado com o qual o Flip-Flop trabalhará e na outra o clock que nos dará o sincronismo do circuito. Temos que ressaltar que, embora o circuito seja multi-valores e o dado seja em três valores, o clock neste caso tem que ser necessariamente binário.

Essa característica do clock pode ser facilmente comprovada quando observamos a fila de portas Alfatopo que recebem o sinal do clock: vemos que se o sinal for 0 (devido ao fato do 0 ser o elemento dominante da porta alfa) ele se sobrepõe a qualquer dado que vier da fila superior do circuito; mas devido ao Alfatopo e topo que vem na seqüência o dado que é enviado para a fila inferior será o 2 (o dado será rotacionado duas vezes) que é o elemento neutro do alfa [tabela 1]. Isto produz um isolamento entre a parte superior do circuito e a saída inferior, mantendo o dado anterior à essa transição.

Quando o sinal de clock passa a ser valor 2, observamos que também não é possível a transferência desse sinal de entrada para os elementos inferiores do Flip-Flop. Fazemos então duas suposições para esclarecer melhor o assunto:

- O clock está passando para dois e o dado continua inalterado D.
- O clock está em dois e o dado está passando para X (valor digital qualquer diferente de D)

Obviamente durante o primeiro intervalo quando ocorrer a alteração do clock o circuito irá transmitir o sinal do dado D que já estava armazenado dentro do Flip-Flop ou na

fila inferior, mas durante a segunda, se o dado alterar-se para X quando o clock já estiver em 2 podemos comprovar que nesse momento não ocorrerá transmissão nenhuma desse sinal para a fila inferior devido à lei da absorção:

$$D\alpha[D\beta(D\gamma X)] = D \quad (I.8)$$

Para demonstrarmos que a lei da absorção é válida basta trabalharmos com o primeiro termo da equação 1.8 utilizando as definições dos operadores:

$$D\alpha[D\beta(D\gamma X)] = D\gamma[D\beta(D\alpha X)]$$

$$D=0 \quad 0\alpha[0\beta(0\gamma X)] = 0$$

$$D=1 \quad 1\alpha[1\beta(1\gamma X)] = 1$$

$$D=2 \quad 2\alpha[2\beta(2\gamma X)] = 2$$

Observando a figura 1.3 vemos que na fila superior o dado que está entrando será X e seguindo a segunda suposição vemos que a fila intermediária está enviando nesse instante para a fila superior D, \bar{D} e \underline{D} respectivamente para as portas Alfatopo da esquerda para direita, esse novo dado X será processado através das portas Alfatopo com os dados vindos da fila inferior naquele instante; temos no seguinte resultado algébrico no último Alfatopo da fila superior:

$$\overline{[(D\alpha X)\alpha\bar{D}]\alpha\underline{D}} \quad (I.9)$$

Passemos agora para a manipulação algébrica da expressão acima utilizando os princípios e propriedades da referência [Serran96] desta dissertação:

$$\left[\overline{\overline{(D\alpha X)}_{\alpha D}} \right] \beta D = \left[\overline{(D\alpha X)}_{\gamma D} \right] \beta D = [(D\alpha X)_{\gamma D}] \beta D \quad (I.10)$$

De acordo com a lei da distributividade crescente sabemos que a operação Beta pode ser distribuída em relação ao Gama, o que nos resulta em:

$$D\beta[D_{\gamma}(D\alpha X)] = [D\beta D]_{\gamma}[D\beta(D\alpha X)] = D_{\gamma}[D\beta(D\alpha X)] \quad (I.11)$$

Chegamos a conclusão que nosso circuito obedecerá a mesma regra da lei da absorção da equação I.8.

Vemos após essa demonstração que embora o dado X tenha condições de ser transferido para as filas inferiores quando o clock está em 2, essa transferência não ocorre porque esse dado alterado não chega a sair da fila superior devido a lei da absorção impossibilitando a gravação desse dado no Flip-Flop.

Caso colocássemos um clock em três valores quando esse sinal passasse por 1 teríamos comportamento inadequado, pois ele atuaria sobre o dado que vem da fila superior produzindo um sinal indesejado; o que podemos observar através da tabela abaixo:

D ou \bar{D} ou \underline{D}	CK	Dado enviado à fila inferior (esse dado é rotacionado 2 vezes)	
0	0	2	Manutenção do Dado anterior à transição do clock
1	0	2	
2	0	2	
0	1	2	Inconsistência
1	1	0	
2	1	0	
0	2	2	Gravação do Dado vindo da fila superior
1	2	0	
2	2	1	

Tabela I.4

Devido a grande sensibilidade do circuito podemos observar essas inconsistências quando introduzimos na simulação um sinal de clock que possua um tempo de subida ou descida superior à 100 ns; neste caso há a ocorrência do fato descrito anteriormente. Mostraremos esse fato mais a frente com as simulações.

1.5 Somador Simples de 3 Valores

Mostramos a seguir o processo de determinação do somador ternário utilizando as regras desenvolvidas anteriormente através da utilização da álgebra de Post.[Post20], [Serran96].

1.5.1 Processador Ternário

A tabela abaixo contém os valores do bloco somador simples, ternário [Martins99].

V, S		B		
		0	1	2
A	0	00	01	02
	1	01	02	10
	2	02	10	11

Tabela I.5

1.5.2 Síntese de S(A,B)

Na tabela abaixo temos os valores de S (a soma digital desejada)

S		B		
		0	1	2
A	0	0	1	2
	1	1	2	0
	2	2	0	1

Tabela I.6

Por simples inspeção concluímos que as linhas são os valores de A deslocados, bem como as colunas são os de B. Através do método de funções auxiliares e escolhendo A como função básica, temos:

$$A_0 = B\beta(A\alpha 1) \quad (I.12)$$

Para as duas outras funções podemos analogamente escrever:

$$A_1 = \bar{B}\beta(\underline{A}\alpha 1) \quad (I.13)$$

$$A_2 = \underline{B}\beta(\bar{A}\alpha 1) \quad (I.14)$$

Para critério de ilustração, transcrevemos na tabela abaixo os valores de A_0 , A_1 e A_2 .

A_0, A_1, A_2		B		
		0	1	2
A	0	0, 1, 1	1, 1, 1	2, 1, 1
	1	1, 1, 1	1, 2, 1	1, 0, 1
	2	1, 1, 2	1, 1, 0	1, 1, 1

Tabela I.7

Por inspeção torna-se evidente que:

$$S = A_0 \gamma A_1 \gamma A_2 \quad (I.15)$$

1.5.3 Síntese da Função V de Transporte

A tabela I.8 apresenta os valores de $V(A,B)$ valores estes que nos indicam quando nosso somador passará a informação de que se pode acrescentar um dígito na soma para o módulo seguinte da soma.

V		B		
		0	1	2
A	0	0	0	0
	1	0	0	1
	2	0	1	1

Tabela I.8

Como anteriormente, utilizamos o método de síntese de funções para chegarmos a expressão da função de transporte.

$$V = A\alpha B\alpha(\underline{A}\underline{\beta}\underline{B}) \quad (\text{I.16})$$

Utilizando-se os teoremas de simplificação mostrados anteriormente podemos mostrar que as equações I.12 a I.16 são equivalentes às equações I.17 a I.21 mostradas abaixo. As figuras 1.4 até 1.8 são os diagramas de blocos dos circuitos finais.

$$A_0 = \underline{\underline{B}}\alpha(\underline{\underline{A}}\alpha\underline{\underline{1}}) \quad (\text{I.17})$$

$$A_1 = \underline{\underline{B}}\alpha(\underline{\underline{A}}\alpha\underline{\underline{1}}) \quad (\text{I.18})$$

$$A_2 = \underline{\underline{B}}\alpha(\underline{\underline{A}}\alpha\underline{\underline{1}}) \quad (\text{I.19})$$

$$S = \underline{\underline{A}}_0\alpha\underline{\underline{A}}_1\alpha\underline{\underline{A}}_2 \quad (\text{I.20})$$

$$V = \underline{\underline{A}}\alpha\underline{\underline{B}}\alpha(\underline{\underline{A}}\alpha\underline{\underline{B}}) \quad (\text{I.21})$$

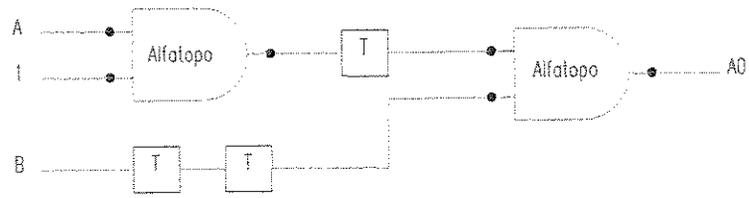


Figura 1.4

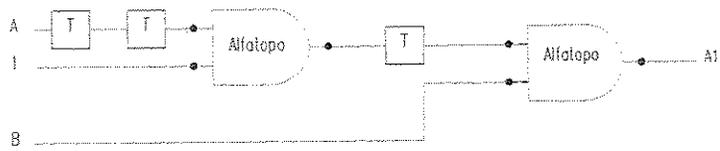


Figura 1.5

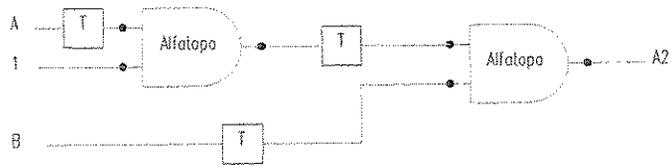


Figura 1.6

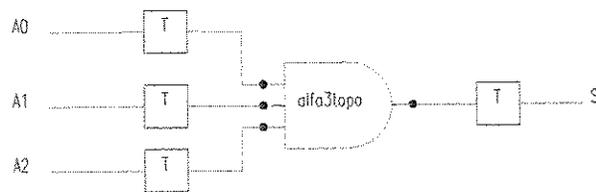


Figura 1.7

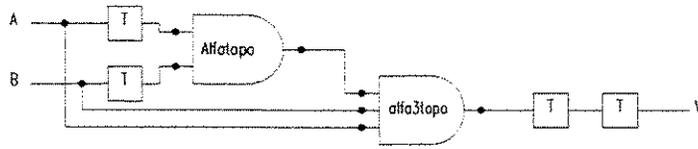


Figura 1.8

1.5.4 Síntese de funções através do Método dos Operadores

Neste item mostraremos a determinação da função Soma do item anterior através do método dos operadores; este foi citado no início desta dissertação como básico para o desenvolvimento de circuitos em lógica multi-valor. O método dos operadores compreende a utilização das funções auxiliares e das tabelas auxiliares para a determinação da função final [Martins99, Martins00].

Através da tabela da função S (para o caso específico das funções ternárias) determinaremos as tabelas auxiliares das funções S_0 e S_1 e mostramos que:

$$S = S_0 \gamma S_1 \quad (I.22)$$

S	B			
	0	1	2	
A	0	0	1	2
	1	1	2	0
	2	2	0	1

Tabela I.9

Para determinarmos a matriz S_0 tomamos S e substituímos os elementos 2 por elementos indiferentes que chamaremos de X; determinaremos os valores de A e B que operados por β resultarão nos elementos 0 da tabela S_0 . A sua função correspondente será determinada pela operação desses elementos obtidos por α .

S_0	B			
	0	1	2	
A	0	0	1	X
	1	1	X	0
	2	X	0	1

Tabela I.10

A tabela acima leva a seguinte função:

$$S_0 = (A\beta B)\alpha(\underline{A}\beta\bar{B})\alpha(\bar{A}\beta\underline{B}) \quad (I.23)$$

Para determinarmos a S_1 tomamos a S e substituímos os elementos 0 por elementos 1, determinaremos os valores de A e B que operados por γ resultarão nos elementos 1 da tabela S_1 ; a sua função correspondente será determinada pela operação, desses elementos obtidos, por β .

S_1		B		
		0	1	2
A	0	1	1	2
	1	1	2	1
	2	2	1	1

Tabela I.11

A tabela nos leva a seguinte função:

$$S_1 = (\overline{A}\gamma\overline{B})\beta(\overline{A}\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma B)\beta(\underline{A}\gamma B)\beta(\underline{A}\gamma B) \quad (I.24)$$

Esta expressão pode ser simplificada utilizando os teoremas das Álgebra de Post, colocando os elementos repetidos em evidência fica:

$$S_1 = [\overline{A}\gamma(\overline{B}\beta B)]\beta[A\gamma(\overline{B}\beta B)]\beta[\underline{A}\gamma(B\beta B)] \quad (I.25)$$

Como foi colocado na equação Y33 obtemos a função S que não pode ser reduzida através dos mesmos teoremas, representada abaixo:

$$S = \{(A\beta B)\alpha(\underline{A}\beta\overline{B})\alpha(\underline{A}\beta B)\}_{\gamma} \{[\overline{A}\gamma(\overline{B}\beta B)]\beta[A\gamma(\overline{B}\beta B)]\beta[\underline{A}\gamma(B\beta B)]\} \quad (I.26)$$

Podemos agora comparar [Nascimento00] as expressões obtidas pelos dois métodos e verificamos que o método dos Operadores nos garante a obtenção de todas as funções ternárias e mecaniza esse processo, mas verificamos que essas funções obtidas são muitas vezes maiores que as obtidas pelo método da inspeção.

Neste caso o método da inspeção nos garantiu uma economia de portas lógicas, que representou um ganho de área em circuito integrado. Podemos comparar a equação I.26 com a equação I.27 (obtida anteriormente através de inspeção) abaixo representada:

$$S = [B\beta(A\alpha 1)]\gamma[\bar{B}\beta(\underline{A}\alpha 1)]\gamma[B\beta(\bar{A}\alpha 1)] \quad (I.27)$$

Capítulo 2

2 - Circuitos Simulados/Resultados Esperados

Neste capítulo serão mostrados as simulações dos circuitos que foram construídos, para que se possam comparar o comportamento da simulação e os dados que foram coletados (esses dados estão presentes no capítulo 4 dessa dissertação) através do circuito que foi implementado.

Todas as formas de onda neste capítulo estão representadas na forma de tensão; pelo fato das correntes possuírem valores muito baixos, optamos pela não representação desses sinais pois o fato das correntes serem muito pequenas é uma característica de circuitos digitais que utilizam elementos CMOS. Durante o curso deste trabalho nós não nos prendemos muito à otimização desses circuitos com relação à frequência, mas à sua funcionalidade e viabilidade de implementação.

Todos os circuitos foram simulados em Spice 6.0 [Spice95] utilizando-se dos modelos em nível 3 da “foundry”, mas alguns problemas observados na simulação do Shift Register nos levaram a utilização do Spice 8.0, pois esse é o único que aceita os modelos em nível 7 que nos permitiram fazer uma simulação mais realista do circuito.

2.1 Alfatopo.

A figura 2.1 é a simulação do circuito Alfatopo mostrado no capítulo anterior onde podemos observar que corresponde à tabela da verdade da função Alfatopo; a parte superior do gráfico é a saída e as entradas encontram-se na parte inferior.

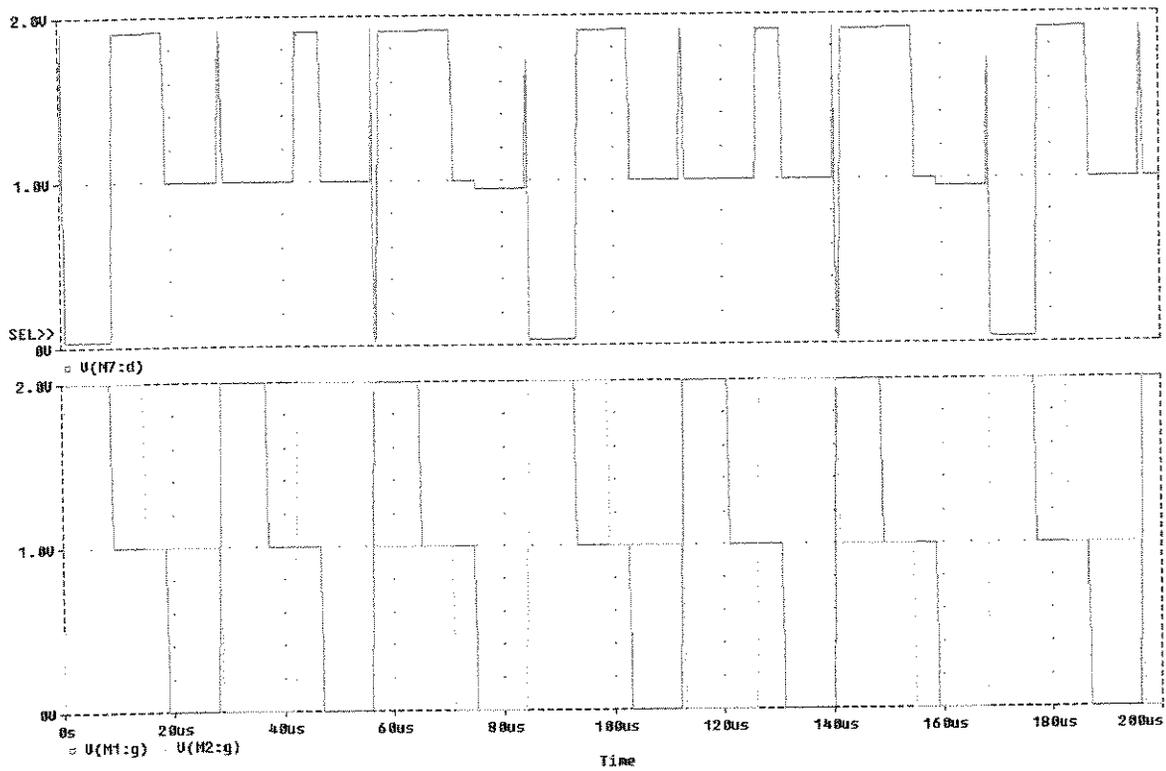


Figura 2.1 :Porta Alfatopo

		A		
		0	1	2
B	$\bar{\alpha}$	0	1	2
	0	1	1	1
	1	1	2	2
2	1	2	0	

Tabela 2.1:Mapa da função Alfatopo

Os “spikes” que podem ser observados na simulação ocorrem quando há a transição dos sinais de entrada; esta é uma interferência indesejada que não pudemos suprimir das simulações sem detrimento do sinal final, neste caso o que melhor se apresentou como um sinal digital ternário. Devemos destacar que esses spikes são de origem numérica, ou seja, não existem no circuito real (conforme será mostrado no capítulo 4).

Na figura 2.2 temos o resultado da simulação do circuito Topo que efetua a negação cíclica de Post ou rotação horária; vemos que ele apenas desloca o sinal de entrada de 1 (um) valor lógico no sentido horário. Na parte superior da figura colocamos a saída isolada, para que possamos novamente observar os “spikes”; na inferior temos os dois sinais juntos onde fica claro o deslocamento que a porta produz.

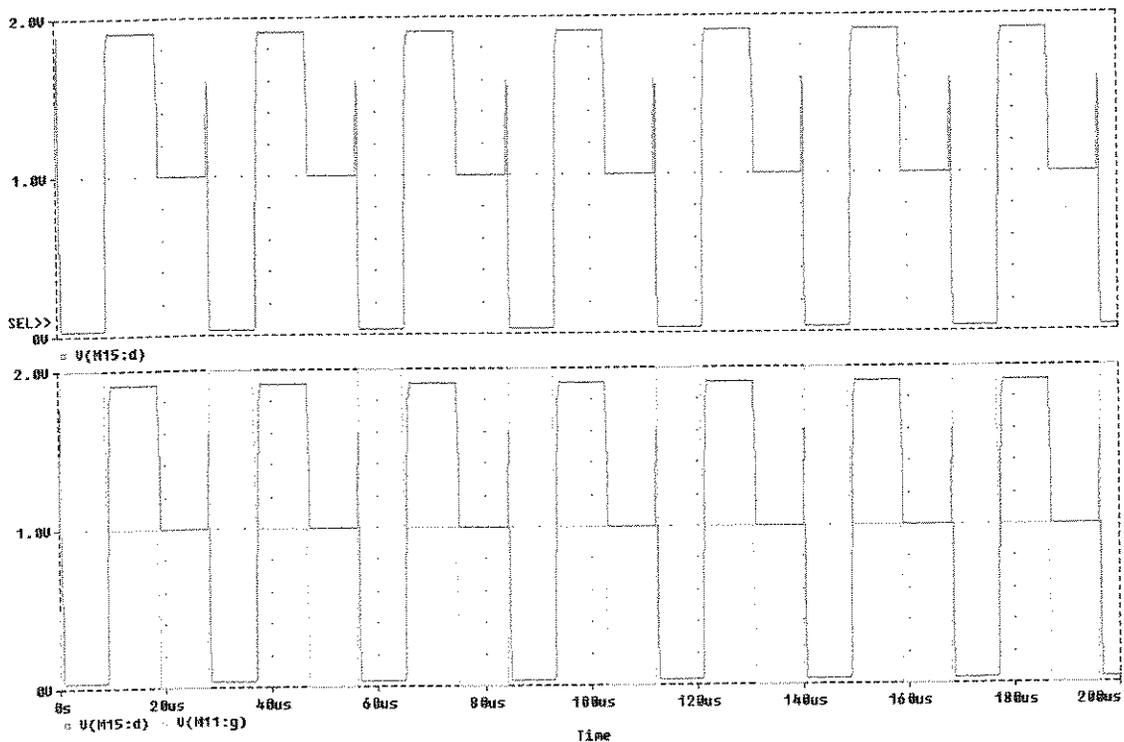


Figura 2.2: Porta Topo.

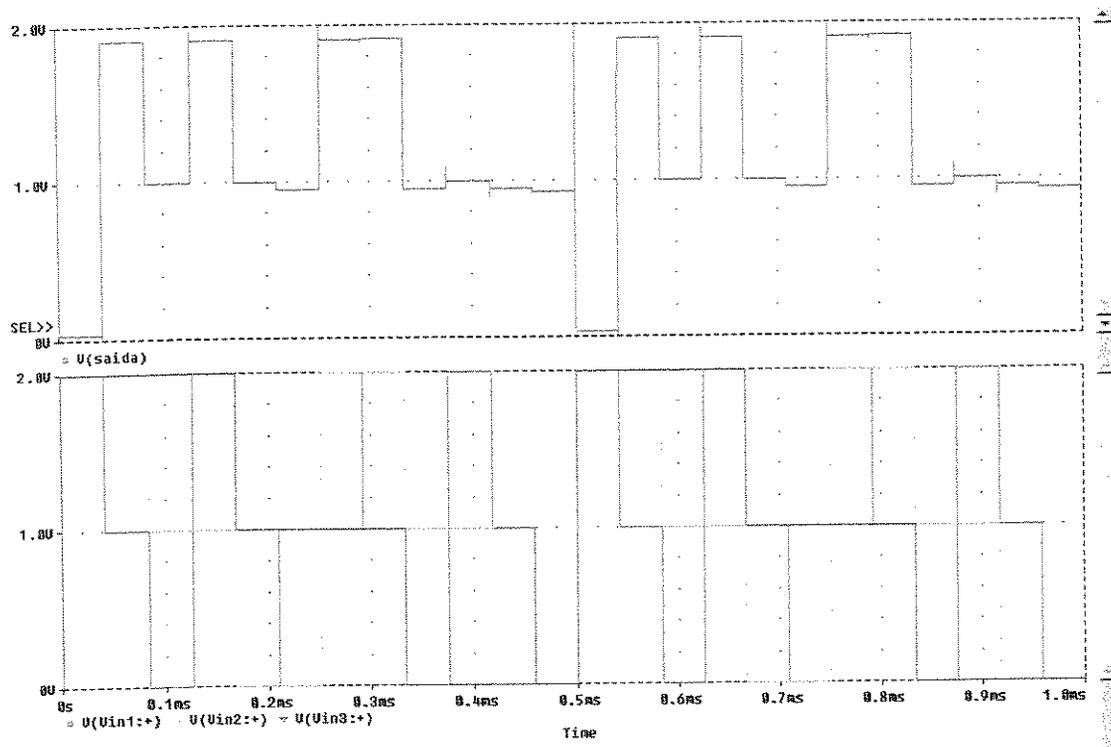


Figura 2.3: Porta Alfa3topo.

Vemos na figura 2.3 o resultados de uma porta Alfa3topo de três entradas que convencionamos chamar de Alfa3topo; para melhorar a visualização, colocamos a saída na parte superior, separada das entradas.

Temos que citar que as tensões de polarização utilizadas nos três circuitos acima se mantiveram exatamente as mesmas, ou seja, V_{pol1} e V_{pol2} são iguais a 1V e 0,8V respectivamente em todos os circuitos e em todas as simulações, inclusive quando mudamos do modelo nível 3 para o nível 7 no Spice 8.0. Observamos no entanto que esses modelos não são totalmente confiáveis e nem representam fielmente a realidade dos componentes.

2.2 Flip-Flop

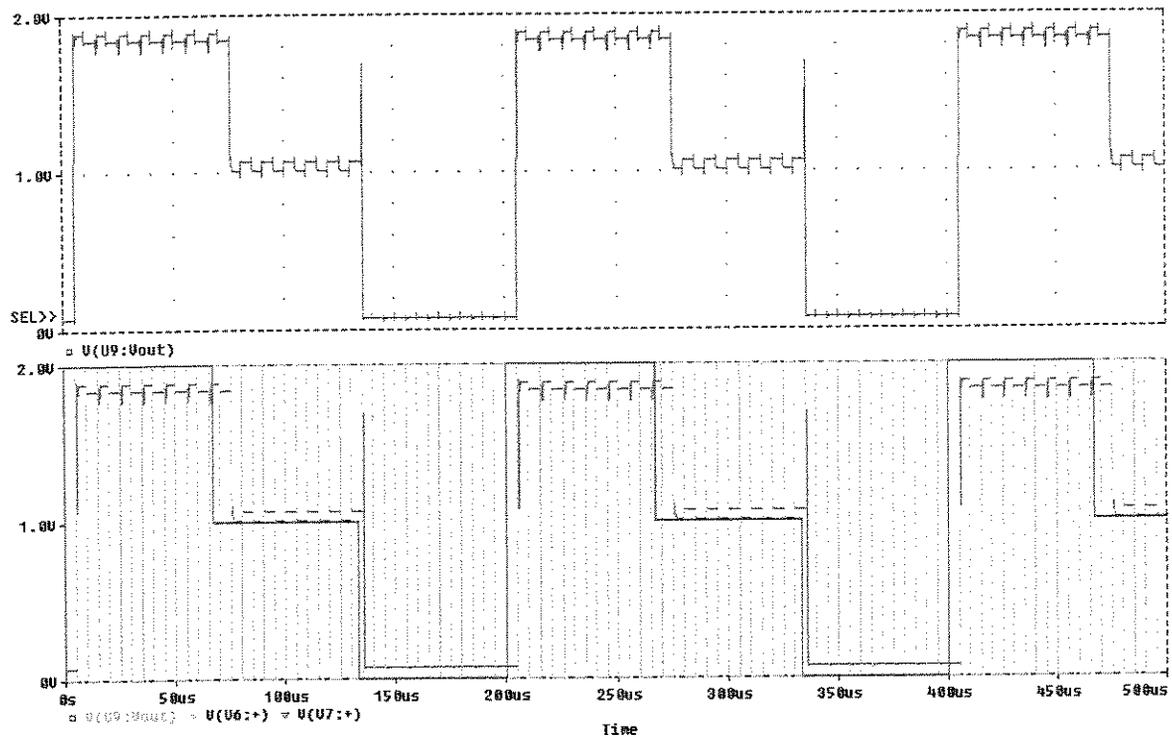


Figura 2.4 :Flip-Flop multi-valor

Na figura acima observamos os três sinais de interesse no Flip-Flop multi-valor, a entrada (vermelho), a saída (azul), e o sinal de clock (verde). Podemos observar que o circuito consegue cumprir a função básica desejada, que é o deslocamento do sinal inicial em um pulso de clock.

Pudemos observar durante essa parte das simulações algumas inconsistências quando introduzimos na simulação um sinal de clock que possua um tempo de subida ou descida superior à 60ns; isto se deve justamente ao fato da impossibilidade da utilização de um clock ternário, fato que já foi comentado nessa dissertação no capítulo 1. A figura 2.5

mostra a ocorrência dessas inconsistências quando introduzimos esses tempos de subida e descida no mesmo circuito que gerou a figura 2.4.

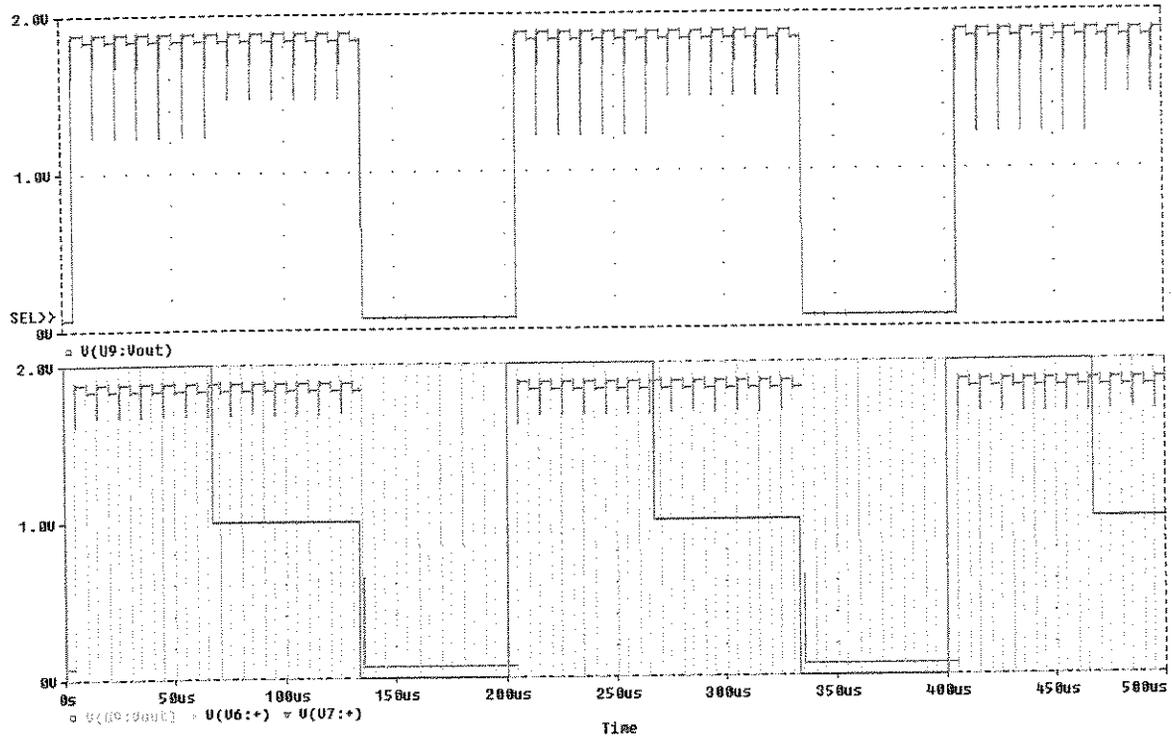


Figura 2.5: Flip-Flop com 0.5us de tempo de subida e descida do clock

A simulação vem confirmar o que foi previsto na parte teórica desta dissertação; quando o tempo de subida ou descida do Clock é muito grande, na comutação do clock de 0 para 2 este sinal passará obrigatoriamente por 1; se esta passagem possuir o Rise Time superior ao mencionado anteriormente o circuito “entenderá” o sinal como possuindo um “1” e processará dessa forma, distorcendo a saída desejada, produzindo as inconsistências mostradas na tabela I.4 do capítulo 1.

2.3 Flip-Flop funcionando como Toggle Switch.

Caso alimentemos a entrada de dados com a saída Topo do Flip-Flop, como mostrado na figura 2.6, iremos obter um multi-vibrador triestável que nos gerará um sinal ternário na saída deste circuito com a frequência de 1/3 do clock original.

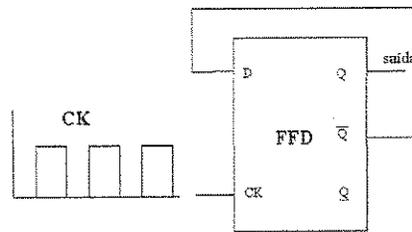


Figura 2.6: Esquema de montagem do Toggle Switch.

O resultado deste circuito encontra-se na figura 2.7 onde confirmamos o funcionamento do circuito como descrito anteriormente.

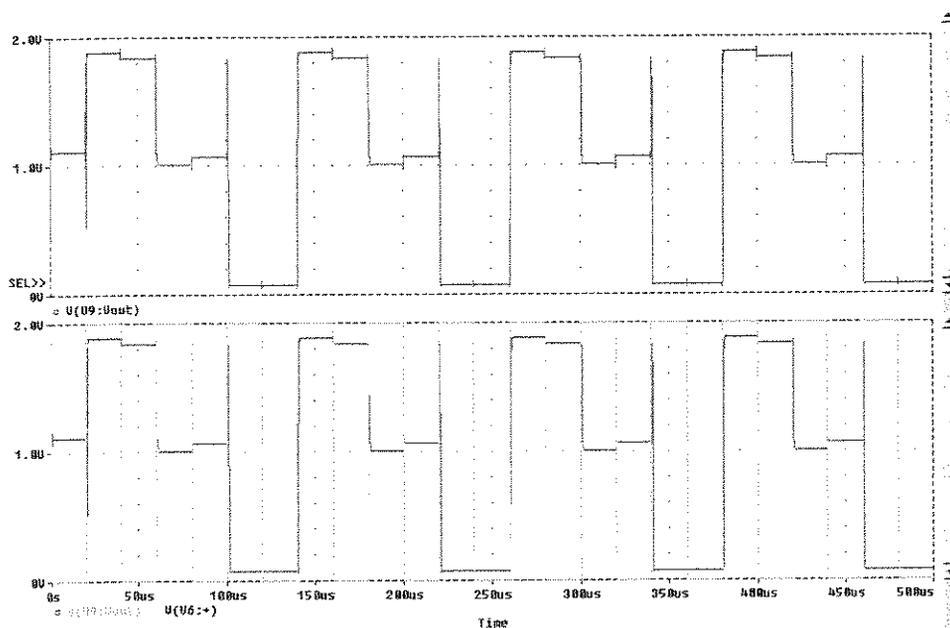


Figura 2.7: Saída do Toggle Switch.

2.4 Quatro Flip-Flops colocados como Shift Register Ternário

Certos sinais internos de um computador necessitam ser atrasados de alguns ciclos do sinal de clock; em algumas aplicações os Flip-Flops são utilizados para produzir esse atraso no sinal original para ser utilizado nos circuitos subsequentes [Tocci98].

Nosso Flip-Flop executa essa função e foi ligado como um elemento binário comum não havendo necessidade do acréscimo de nenhum outro circuito para a obtenção da mesma. Na figura 2.8 podemos observar como foi ligado e o resultado da simulação na figura 2.9.

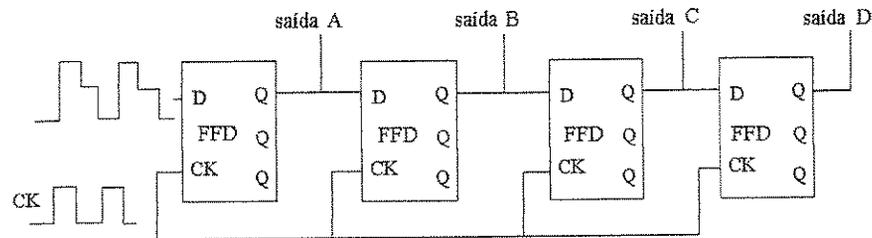


Figura 2.8: Esquema de ligações do Shift Register.

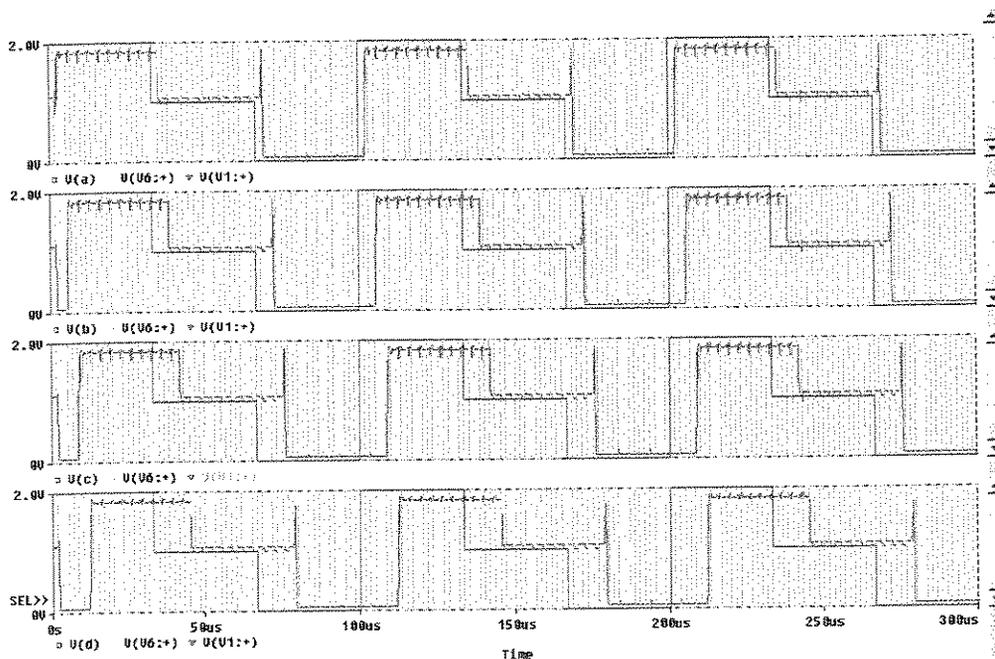


Figura 2.9: Resultado da Simulação.

2.5 Somador

Podemos observar o resultado da simulação do circuito Somador Simples na figura 2.10; vemos respectivamente o resultado da soma, o resultado do circuito de Carry e os sinais de entrada.

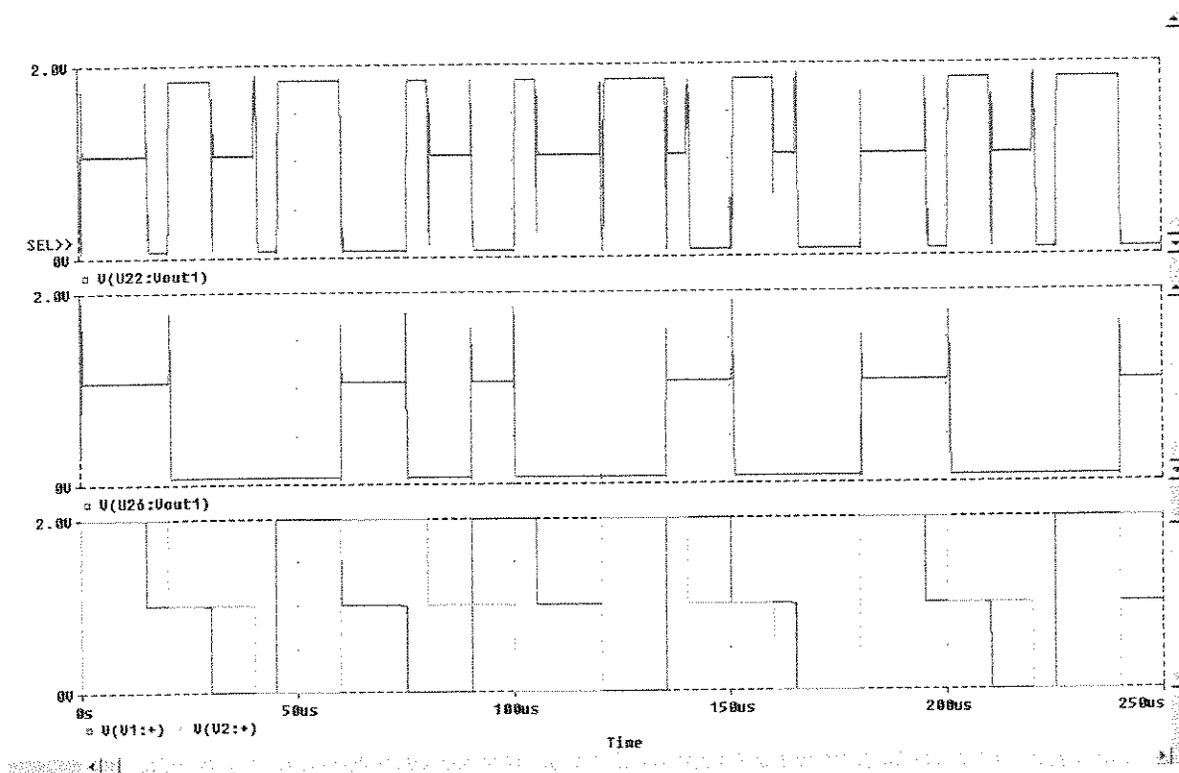


Figura 2.10: Meio Somador Ternário.

Optamos pela montagem do meio somador visando uma economia de área no chip; como sabemos, pode-se montar em lógica binária o somador completo através da utilização de dois meio somadores [Tocci98]; para a lógica ternária isto continua valendo, com a diferença de utilizarmos uma porta Beta no lugar da porta NOR do binário. A partir do que foi dito necessitamos apenas da utilização de dois chips para efetuarmos a montagem (a

AMS nos fornece cinco chips encapsulados pedido); assim poderemos efetuar a montagem com os elementos que nos serão fornecidos, a montagem sendo igual a da figura 2.11.

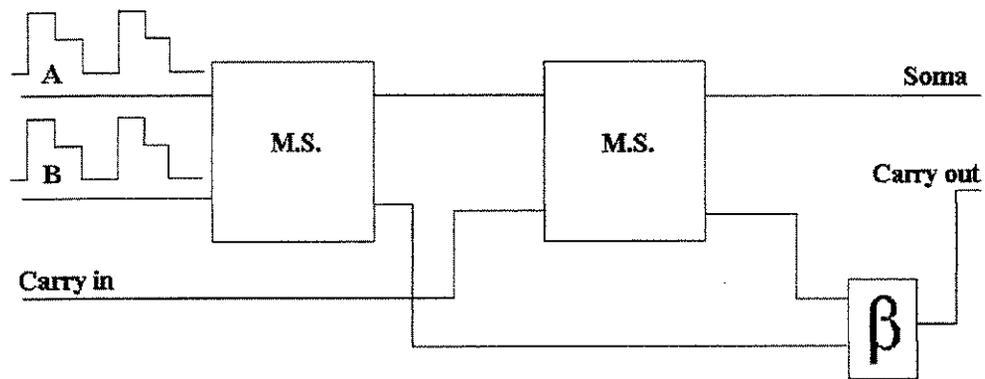


Figura 2.11: Esquema do Somador Completo Ternário.

Capítulo 3.

3.1 Circuito Integrado Fapesp73.Dem (Lay-Out e Sistemas de Medidas)

Após a fase de simulação, todo circuito integrado precisa passar pela confecção do seu Lay-Out; esta etapa pode ser descrita como o desenho das diversas máscaras que serão usadas na gravação da geometria final do circuito sobre a lâmina de silício.

Para o desenvolvimento do Lay-Out utilizamos o programa para PC L'Edit da Tanner [Tanner96]. Existem muitas outras opções, tanto em PC's como em workstations, mas esta mostrou-se a mais viável. Como outra opção para PC temos o Winlasi, que é distribuído gratuitamente via Internet. Para Workstation temos o IC Design da Mentor e o Cadence; o pacote da Mentor é didaticamente o melhor porque a maioria dos designers usam, mas ele precisa de uma biblioteca da foundry que não temos em nossa versão, impossibilitando assim sua utilização.

As estruturas que planejamos colocar no C.I. foram:

- 1 Transistor canal N
- 1 Transistor canal P
- 1 Porta Topo
- 1 Porta Alfatopo
- 3 Flip-Flops tipo D
- 3 Meios Somadores

Caso nossas estruturas apresentassem um comportamento diferente do esperado, através do acesso aos transistores teríamos meios de determinar se o processo de confecção do circuito integrado foi efetuado seguindo os parâmetros do modelo que a foundry distribui. Assim sendo, se as tensões de limiar dos transistores fossem muito divergentes das que estão nos modelos ou se suas curvas características apresentassem grandes distorções isso nos mostraria as possíveis causas daquelas dispersões.

Observamos pequenas diferenças nas curvas dos transistores entre os C.I's; todas as tensões de limiar dos transistores encontravam-se na faixa de valores do modelo da foundry, o que mostra que o processo da AMS encontra-se bastante estável e seguro para aplicações em geral.

Decidimos também colocar as portas básicas para determinarmos algum outro desvio nas funções, caso alguma estrutura mostrasse um comportamento diferente do esperado.

A inclusão dos três Flip-Flops foi motivada pela possibilidade de obter montagens mais complexas como o shift register. Pretendíamos também montar um somador completo utilizando os meio somadores e as outras portas básicas.

3.2 Montagem Experimental

Para executarmos as medidas do circuito integrado que foi confeccionado utilizamos as instalações do Laboratório de Medidas do Departamento de Microeletrônica; nestas instalações contamos com os equipamentos essenciais para a devida execução dessas medidas.

Entre os equipamentos utilizados estão:

Osciloscópio HP 54503A – Digital.

Quatro canais.

Largura de Banda de 500Mhz.

Entrada para barramento GPIB.

Gerador de Forma de Onda Arbitrário – 33120A.

Frequência Máxima de Operação - 15 Mhz.

Geração Digital de formas de Onda.

Fonte Estabilizada E 3631 A.

Tripla Saída DC.

0 – 6V, 5A / 0 - ±25V, 1A.

Como pode-se observar na figura 3.1 os instrumentos utilizados foram controlados através do GPIB e alguns sinais foram gerados na placa LAB-PC 1200.

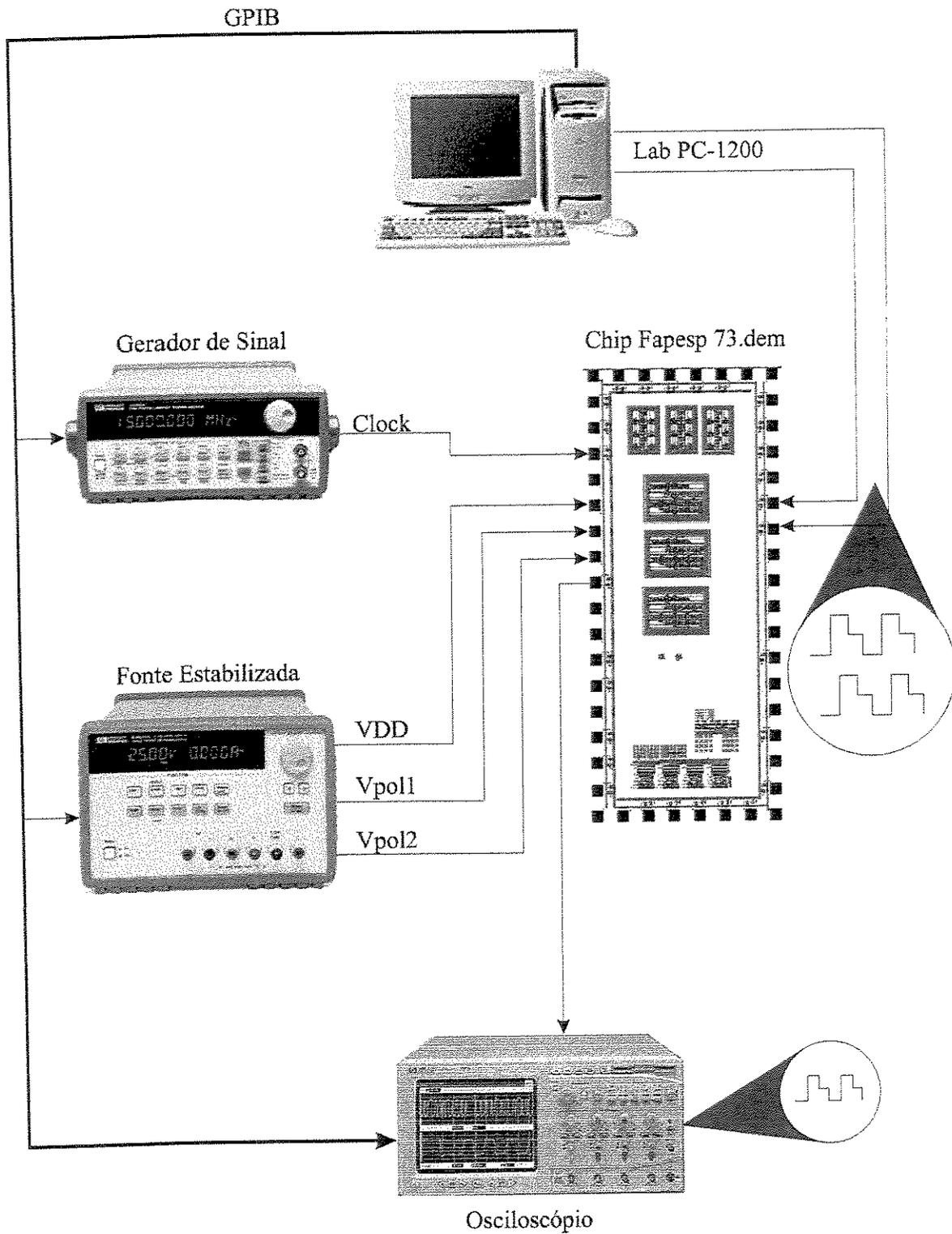


Figura 3.1

3.3 GPIB

Em 1965 Hewlett Packard lançou o Hewlett Packard Interface Bus (HPIB) para conectar sua linha de instrumentos programáveis a seus computadores. Por causa das suas elevadas taxas de transferência (1 Mbps) esta interface ganhou rapidamente popularidade. Mais tarde foi aceito como o padrão IEEE 488-1975, e evoluiu para o padrão ANSI/IEEE 488.1-1987. Hoje em dia chamamos de General Purpose Interface Bus (GPIB). O padrão ANSI/IEEE 488.2-1987 fortaleceu o padrão original, definindo precisamente como controlar e comunicar instrumentos.

A função principal do GPIB é a troca de informação entre um ou mais dispositivos. Antes que qualquer dado seja enviado, os dispositivos devem ser configurados para enviar informação de acordo com o respectivo protocolo.

O controlador de sistema de GPIB é chamado moderador. Como numa discussão (neste caso a transferência de dados) ele (o moderador) controla quem deve atuar na altura certa. É da responsabilidade do moderador garantir que não haja mais de um dispositivo enviando dados ao mesmo tempo, e qual desses dispositivos tem autorização para o envio dos dados. O moderador também determina qual dispositivo estará esperando a informação (escuta) quando a informação não se destina a todos os dispositivos. Cada vez que há envio de dados, existe uma prévia verificação de que os dados irão ser recebidos (o chamado *hand-shaking*).

Esta forma de operar faz com que a velocidade da transmissão dependa da velocidade dos dispositivos, constituindo uma desvantagem. Exemplificando, em 10

dispositivos, se 9 funcionarem a 5 Mbytes por segundo e apenas 1 funcionar a 1 Mbyte por segundo, a velocidade de transmissão será de 1 Mbyte por segundo.

Cada dispositivo num sistema GPIB tem um endereço único, que varia entre 0 e 30. Esse endereço serve de identificação para cada dispositivo. No processo de transferência de informação, o uso do endereço é vital e necessário, pois dessa forma só o dispositivo especificado (através do seu endereço) receberá a informação transmitida.

Os dispositivos GPIB se comunicam com outros dispositivos GPIB enviando dois tipos distintos de mensagens:

- Mensagens Dependentes do Dispositivo: freqüentemente chamados dados ou mensagens de dados, contém informação específica do dispositivo, tal como: instruções de programação, resultados de medidas, estados da máquina, etc.
- Mensagens de Interface: Usualmente chamados comandos ou mensagens de comando, são os que realizam funções como: inicializar o barramento, direcionar dispositivos e setar modos para programação local ou remota [National I.99].

3.3.1 Tipos de dispositivos GPIB:

Há três tipos de dispositivos GPIB: *talkers*, *listeners* e/ou *controllers* (chamador, ouvinte e controlador, respectivamente). Um *talker* envia mensagens de dados a um ou mais *listeners*, os quais recebem os dados. O controlador lida com o fluxo de informação mediante o envio de comandos a todos os dispositivos. Um voltímetro digital, por exemplo, é um *talker* e também um *listener*.

A função do GPIB *controller*, é comparável à de uma CPU de um computador, mas a melhor analogia é compará-lo com uma central de comutação de um sistema telefônico. O *controller* monitora o barramento GPIB e quando recebe o aviso de que um dispositivo necessita enviar uma mensagem de dados a outro, conecta o *talker* com o *listener*. O *controller* usualmente direciona e ativa o *talker* e o *listener* antes que o *talker* possa enviar sua mensagem ao *listener*. Depois que a mensagem é transmitida, o *controller* pode direcionar outros *talkers* e *listeners*.

Algumas configurações GPIB não requerem um controlador. Por exemplo um dispositivo que sempre é um *talker* (chamado "*talk only device*", dispositivo somente chamador) é conectado a um ou mais "*listen only device*". Um controlador é necessário quando o *talker* ou o *listener* podem ser trocados. Geralmente a função de *controller* é realizada por um computador (como em nosso caso). Não obstante podem haver vários *controllers* em um barramento GPIB, porém somente podem ser ativos um por vez, chamado controlador a cargo (CIC). O controle ativo pode passar do CIC atual para outro que se encontra em estado de espera [National I.99].

3.3.2 Linhas e sinais GPIB:

A interface GPIB consiste de 16 linhas de sinal, agrupadas em 8 de dados, 3 de *handshake* e 5 de gerenciamento de interface e 8 linhas de terra (ver figura 3.2)

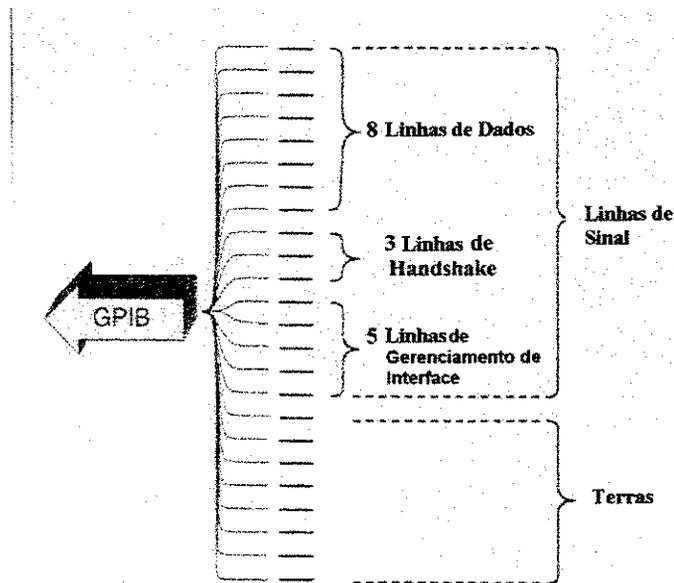


Figura 3.2

Os tipos de linhas são:

a) *Linhas de Dados*: As 8 linhas de dados, desde DIO1 até DIO8 levam tanto mensagens de dados como mensagens de comando. Todos os comandos e muitos dos dados utilizam o código ASCII de 7 bits, e neste caso o oitavo bit (DIO8) fica sem uso e utiliza-se para paridade.

b) *Linhas de Handshake*: 3 linhas controlam assincronamente a transferência de mensagens entre dispositivo. Este processo se chama "*3 wire interlocked handshake*" (acordo de 3 vias) e garante que as mensagens nas linhas de dados sejam enviadas e recebidas sem erros de transmissão.

As funções de cada linha são:

NRFD (*not ready for data*): indica quando um dispositivo está pronto ou não para receber uma mensagem.

NDAC (*not data accepted*): indica quando um dispositivo aceitou ou não uma mensagem.

DAV (*data valid*): avisa quando os sinais nas linhas de dados estão estáveis e podem ser aceitadas com segurança pelos dispositivos.

c) Linhas de Gerenciamento de Interface: são 5 linhas que gerenciam o fluxo de informação através da interface.

Cada uma delas tem a seguinte função:

ATN (*attention*): quando está no estado alto usa as linhas de dados para enviar comandos, e quando está em baixo usa as linhas de dados para enviar dados.

IFC (*interface clear*): o controlador do sistema gerencia a linha de IFC para inicializar o barramento e converter-se no CIC.

REN (*remote enable*): o controlador do sistema gerencia a linha REN para colocar os dispositivos nos modos de programação local ou remota.

SRQ (*service request*): qualquer dispositivo pode gerenciar esta linha para solicitar assincronamente um serviço desde o controlador.

EOI (*end or identify*): esta linha tem dois propósitos, o *talker* usa para marcar o final de uma cadeia de mensagens e o *controller* usa para chamar dispositivos para identificar suas respostas paralelamente [National I.99].

3.4 Características físicas e elétricas:

Os dispositivos são geralmente conectados com um cabo blindado de 24 condutores, dispostos tal como se observa na figura 3.3, e podem ser conectados numa configuração linear, ou uma configuração estrela ou uma combinação de ambas, como pode-se ver nas figuras 3.4 e 3.5.

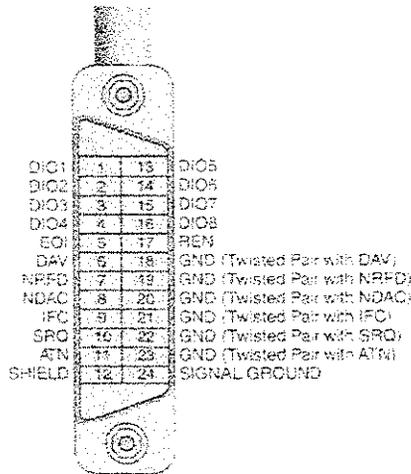


Figura 3.3

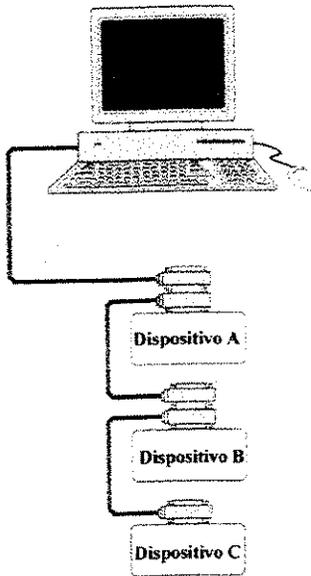


Figura 3.4

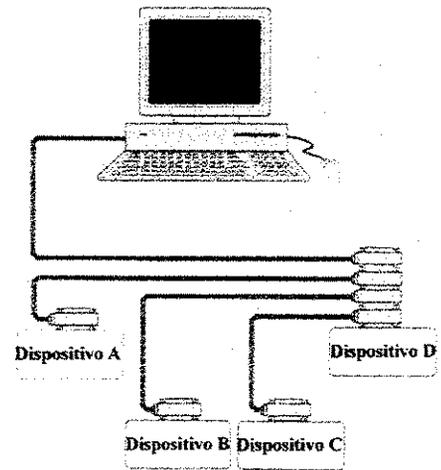


Figura 3.5

Para o caso do nosso laboratório, por limitações físicas dos cabos, utilizamos a configuração da figura 3.4, assim conseguimos atingir todos os equipamentos necessários para nossas experiências.

A interface GPIB usa lógica negativa com níveis TTL padrão, quando uma linha está ativa se corresponde com um nível baixo (menor que 0,8 V) e quando está inativa se corresponde com um nível alto (maior que 2V) [National I.99].

3.5 Requisitos de configuração:

Para conseguir uma elevada taxa de transferência de dados, a distância física entre dispositivos e o número de dispositivos estão limitados. As restrições típicas para uma operação normal são:

- Uma separação máxima de 4 m entre dois dispositivos qualquer e uma separação média de 2 m em todo o barramento.
- Um comprimento total do cabo de no máximo 20 m.
- Não mais do que 15 dispositivos conectados a cada barramento, com não mais de 2/3 ativos a cada vez.

Para trabalhar a maiores velocidades e usando handshake existem maiores restrições.

Observando essas restrições percebemos que nossas instalações estão dentro do desejado para uma elevada taxa de transferência [National I.99].

3.6 IEEE 488.2 e SCPI:

Os padrões IEEE 488.2 e SCPI (*standard commands for programmable instruments*) superam as limitações e ambigüidades do padrão original do IEEE 488.

IEEE 488.2 possibilita desenhar sistemas mais compatíveis e produtivos, e SCPI simplifica a tarefa de programação definindo um conjunto de comandos simples e compreensivos para instrumentos programáveis sem importar qual é o seu fabricante. O padrão IEEE 488 simplificou a interconexão de instrumentos programáveis mediante a definição das especificações mecânicas, elétricas e o protocolo de hardware. Desta forma, instrumentos de diferentes fabricantes podem ser interconectados por um cabo padrão. Este padrão apresentava alguns defeitos, não especificando os formatos dos dados, relatórios de estado, comandos de configuração, etc. Como resultado disto cada fabricante implementou estes aspectos de diferentes maneiras, complicando a tarefa do designer do sistema.

IEEE 488.2 melhorou e fortaleceu a IEEE 488 padronizando formatos de dado, relatórios de estado, gerenciamento de erros e comandos comuns, com os quais todos os instrumentos respondem de maneira definida. Devido a isto os sistemas IEEE 488.2 são muito mais compatíveis e confiáveis.

O padrão SCPI construiu-se sobre o IEEE 488.2, e define comandos específicos dos dispositivos que padronizam a programação de instrumentos. Os sistemas SCPI são muito mais fáceis de programar e manter. Em muitos casos se podem trocar ou atualizar instrumentos sem ter que mudar o programa de teste. A combinação de SCPI e o IEEE 488.2 implica em um ganho substancial na produtividade [National I.99].

3.7 LAB - PC 1200

Utilizamos na placa LAB-PC 1200 as duas saídas dos conversores digital-para-analógico (DAC0OUT e DAC1OUT) para gerar os sinais ternários necessários para testar os elementos do Chip; poderíamos ter gerado esses sinais no Gerador de Forma de Onda Arbitrário – 33120A, mas em certas partes dos testes precisaríamos de um sinal de clock que seria gerado pelo gerador e a placa geraria o sinal ternário necessário.

3.8 Instrumentos Virtuais Utilizados

Os chamados Instrumentos Virtuais (ou V.I's como são conhecidos) são programas desenvolvidos em Labview para desempenhar uma tarefa específica desejada do programador. Labview é um programa de desenvolvimento de aplicações como qualquer outro comercial, como C ou BASIC, mas ao invés de usar textos para criar as linhas de código utiliza uma linguagem de programação gráfica chamada de linguagem G, para criar programas na forma de diagramas de blocos que obedecem a uma estrutura hierárquica.

Esses blocos que são criados agem como uma seqüência de ações para trabalhar com os dados iniciais do problema em questão, estes podem ser a aquisição dados de um osciloscópio ou gerar sinais de controle da tensão de um motor. Cada Instrumento Virtual pode conter vários blocos destes para solucionar problemas específicos, são as chamadas sub-VI's. Essa modularidade torna o Labview um programa extremamente poderoso, pois assim cada bloco que é desenvolvido para uma determinada solução poderá ser utilizado em outro problema como parte da solução necessária.

A mencionada estrutura hierárquica do Labview é de extrema utilidade pois torna a programação uma atividade muito visual e intuitiva, podemos observar utilizando o Labview a seqüência em que ocorre a ação desejada, caso exista algum problema este é facilmente detectado.

Durante a fase de testes do circuito integrado precisávamos desenvolver três instrumentos virtuais que obedecessem às especificações das medidas que seriam efetuadas: um instrumento que fizesse a coleta dos dados no osciloscópio, um que gerasse as formas de onda ternárias na placa DAC - PC 1200 e ao mesmo tempo sincronizasse esse sinal gerado com o clock do gerador de sinais arbitrário (33120 A), e o terceiro para a visualização (com possibilidade de salvar os dados) dos canais do osciloscópio.

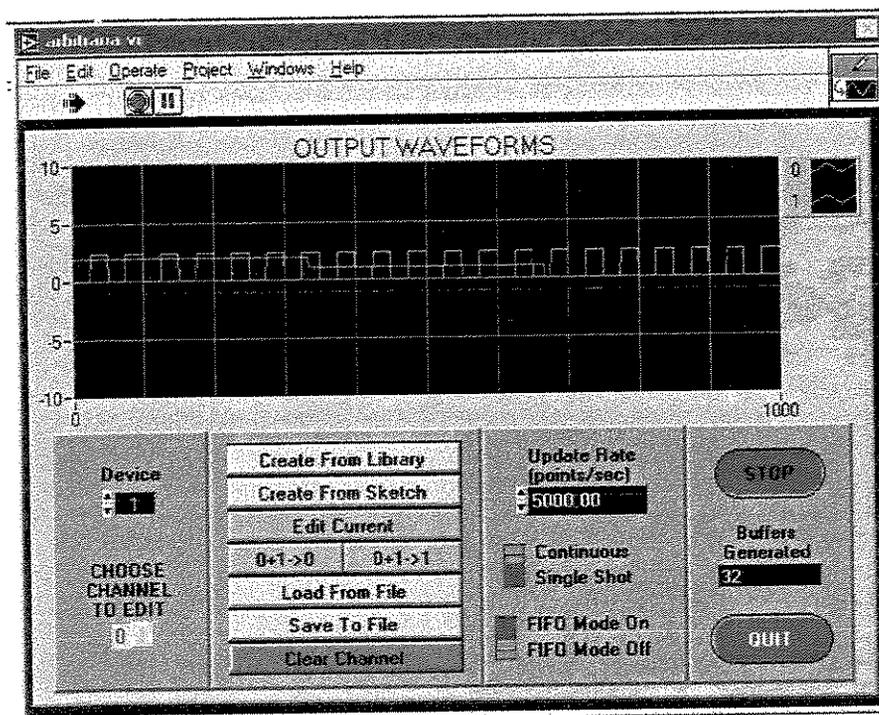


Figura 3.6

Na figura 3.6 vemos o painel frontal do instrumento virtual "arbitraria.vi", esse instrumento tem como função a geração de qualquer forma de onda na saída digital da

placa LAB - PC 1200. Existem duas possibilidades de funcionamento desse instrumento, a primeira seria carregar as formas de onda de um arquivo previamente gerado e a segunda seria desenhar diretamente na tela da V.I. a forma desejada.

Optamos pela geração do arquivo utilizando uma segunda V.I. criada para esse propósito, assim teríamos uma precisão maior nos níveis desejados dos sinais ternários, criamos assim a V.I. geratudo.vi e seu painel principal está na figura 3.7

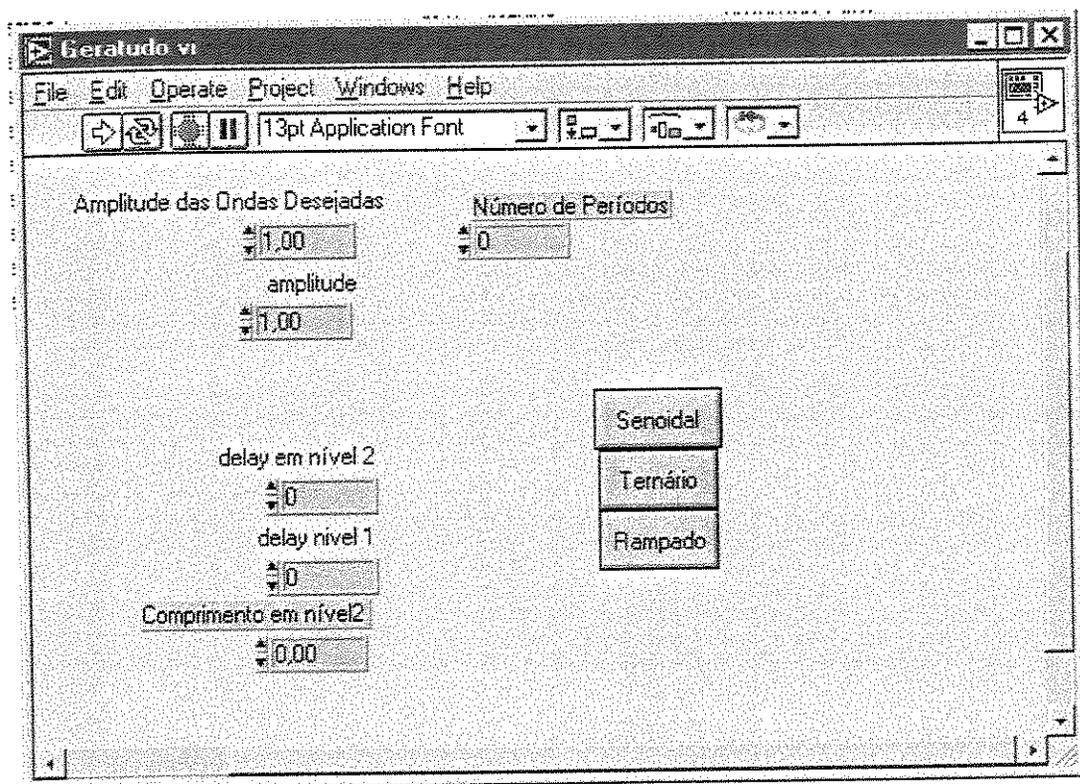


Figura 3.7

Através desse instrumento pudemos criar todas as formas de onda que foram geradas na placa através da arbitrária.vi e que forma necessárias para as medidas do C.I.

Para o sistema de aquisição dos dados do osciloscópio criamos uma V.I. que está mostrado na figura 3.8, este instrumento possibilita a leitura de qualquer canal desejado do osciloscópio.

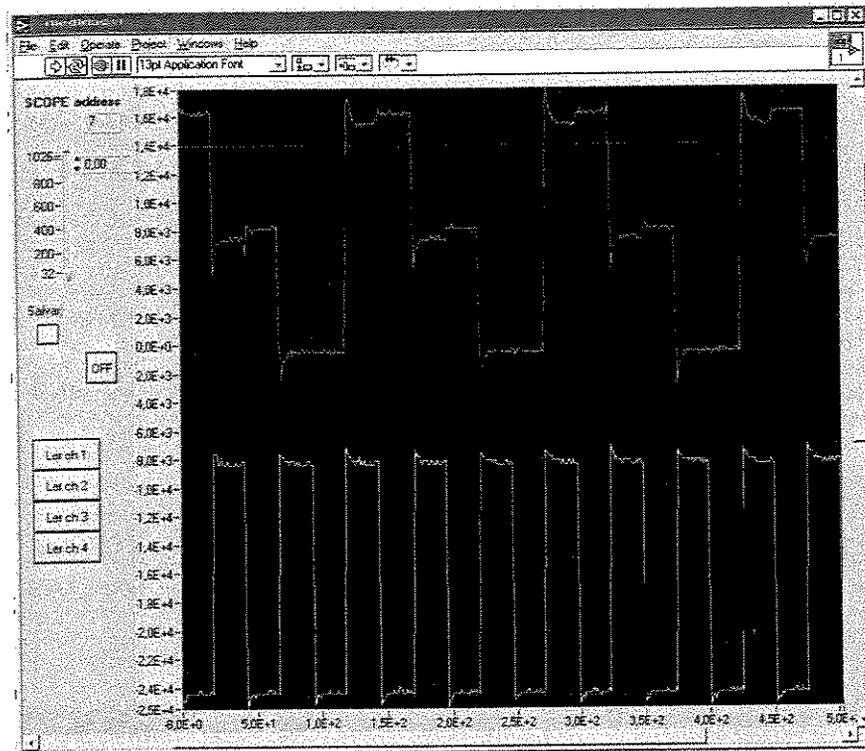


Figura 3.8

Quando os dados observados na tela corresponderem aos desejados existe a opção de salvar as formas de onda em disco para posterior utilização; esses arquivos podem ser facilmente abertos e trabalhados em Excel.

No anexo A está a documentação das V.I.'s; através dela podem-se recriar esses instrumentos caso desejado, o Labview gerando automaticamente essa documentação para a formação de um Back-up físico [Labview96].

Capítulo 4

4 Resultados Obtidos

Através das montagens mostradas no capítulo anterior obtivemos os sinais esperados das simulações; obviamente tivemos alguns desvios e falhas, os motivos de suas ocorrências serão elucidados durante este capítulo.

4.1 Topo

Como mostramos anteriormente a porta Topo funciona como um Shift anti-horário no sinal original, na figura 4.1 podemos observar os sinais de entrada e saída sobrepostos para a melhor visualização deste deslocamento.

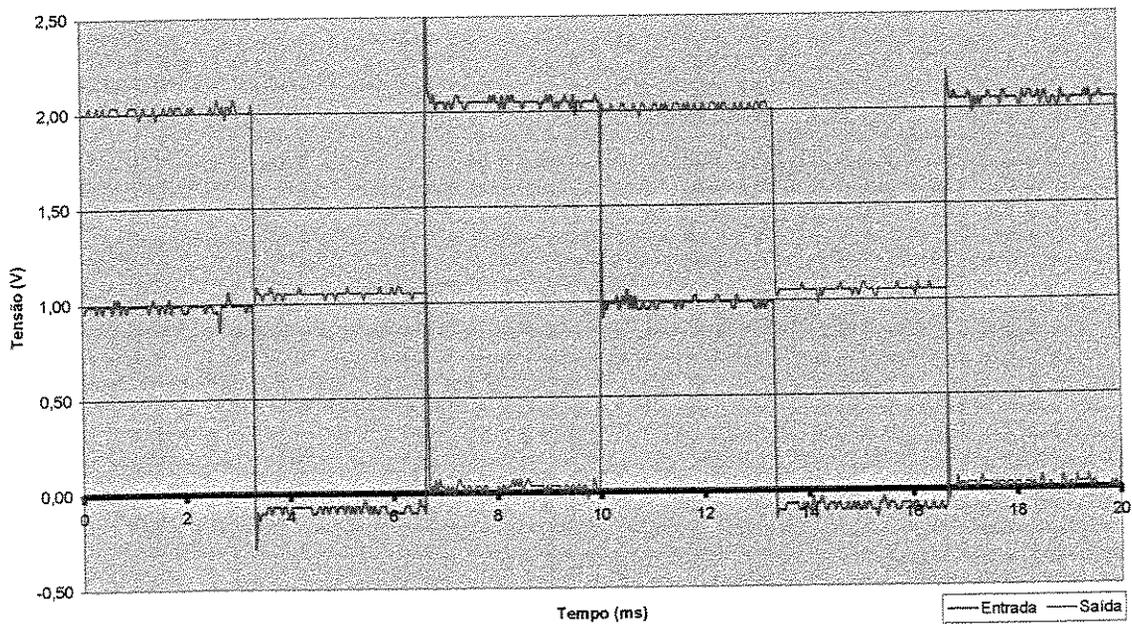


Figura 4.1

Para uma melhor visualização da operação da porta Topo utilizamos uma entrada em rampa onde podemos observar mais claramente a transição dos valores lógicos. Durante

as medidas observamos que esta entrada mostrou-se muito útil para os ajustes das curvas através das tensões de polarização. Essas curvas estão representadas na figura 4.2

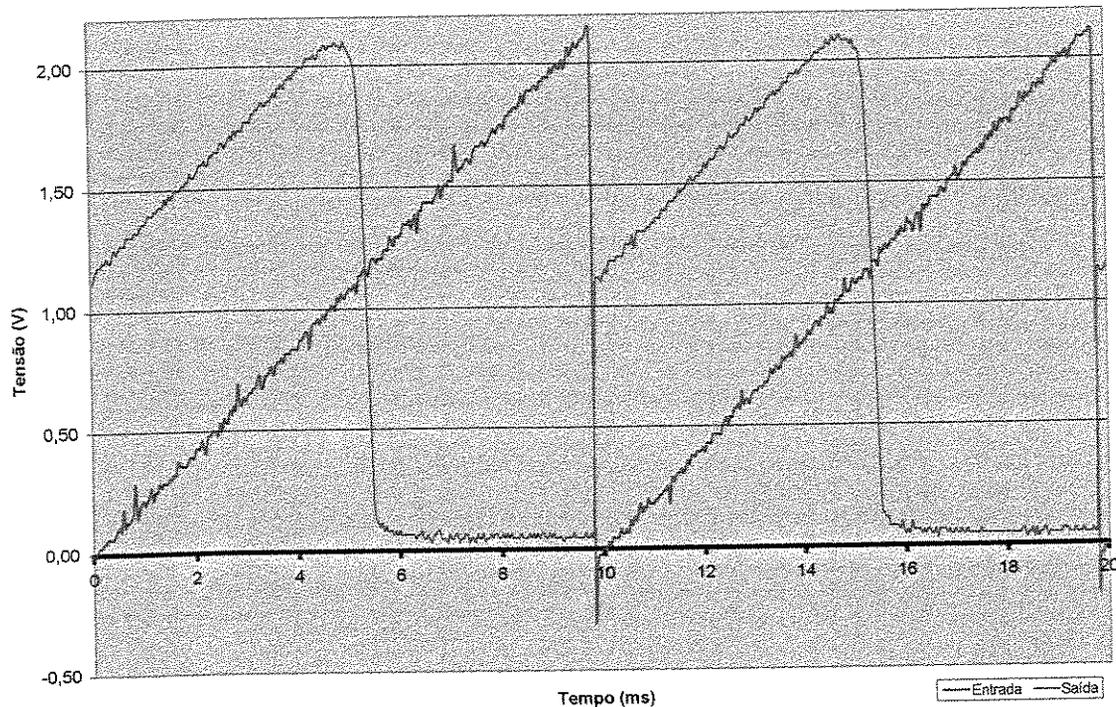


Figura 4.2

Para uma melhor visualização das curvas na figura 4.2 introduzimos um offset no osciloscópio de forma a centrar as curvas na tensão correspondente à 1V (valor lógico "1"). Assim podemos observar todos os três valores lógicos com mais clareza; posteriormente essas curvas foram acertadas, retirando o offset que tinha sido colocado.

Conforme foi mostrado nesta dissertação temos duas tensões com acesso externo para possíveis ajustes; durante a fase de testes houve a necessidade de alteração destas tensões no circuito integrado, incluindo alterações da tensão de alimentação para possibilitar o funcionamento do circuito.

Observamos que a porta Topo funciona perfeitamente dentro de uma faixa de tensões de alimentação e também dentro de uma faixa nas tensões de polarização; posteriormente veremos que esta característica mostrou-se muito útil para o funcionamento de circuitos mais complexos. Essa faixa de tensões de trabalho altera-se também conforme a carga que é colocada na porta. A tabela 4.1 mostra as faixas de funcionamento da porta Topo.

	Faixa de Funcionamento do Circuito	
Tensão de Alimentação	2,3V	2,5V
Tensão de Polarização 1 Vpol1	1,3V	1,5V
Tensão de Polarização 2 Vpol2	0,9V	1,40V

Tabela 4.1

4.2 Flip-Flop

Na fase de testes observamos que o Flip-Flop Multi-Valores tem o comportamento esperado da parte teórica desta dissertação, confirmando assim as simulações que foram desenvolvidas no capítulo 2; temos apenas a ressaltar que houve a necessidade da alteração das tensões de alimentação e das polarizações.

No caso dos Flip-Flops ternários podemos ter três saídas funcionais: a parte que tem como saída o Dado de entrada (D), o dado deslocado em Topo (\overline{D}) e o dado deslocado em Base (\underline{D}). Todos esses elementos estão deslocados de um sinal de clock em relação ao sinal de entrada original.

Para implementarmos o Toggle Switch em nosso circuito tivemos obrigatoriamente que realimentar a entrada com as saídas Topo (\bar{D}) ou Base (\underline{D}), pois somente assim conseguiríamos o nosso oscilador ternário. No caso da saída que nos fornece o dado (D), caso esta seja usada na realimentação, não produzirá oscilação porque, como é de se esperar, ela fixa a saída em um único valor indefinidamente.

Conforme nossas observações durante a fase de simulações, a saída com melhor resposta ao toggle foi a saída Base do Flip-Flop. A saída Topo mostrou-se pouco eficiente, perdendo muitas vezes o sinal original.

Assim durante a construção do circuito integrado escolhemos a saída Base como a única acessível externamente (devido à limitação no número de pinos do encapsulamento do integrado – 48 pinos máximo oferecido pela foundry) assim os resultados obtidos durante as medições são referentes à saída Base do nosso circuito como podemos observar na figura 4.3

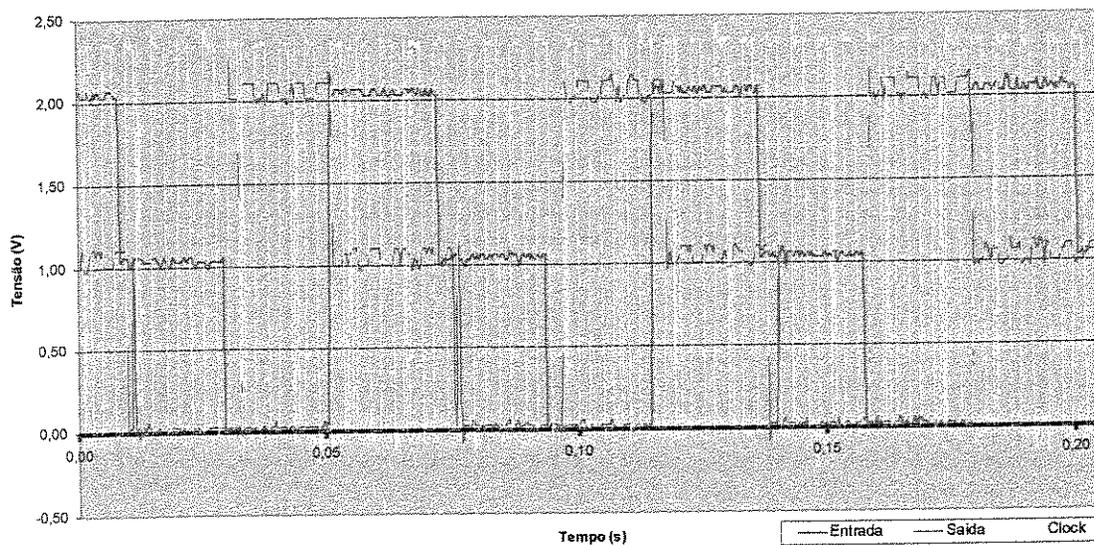


Figura 4.3

Podemos observar na figura 4.3 o atraso de 01 sinal de clock que é esperado em um Flip-Flop tipo D.

Para uma melhor observação do funcionamento do Flip-Flop colocamos na entrada do circuito mais dois tipos de sinais que ilustram claramente seu funcionamento, o sinal em rampa e o sinal em senoide.

Como destacamos anteriormente, a saída que está sendo medida é o sinal base da entrada, observamos claramente o comportamento digital da porta nas figuras 4.4 e 4.5.

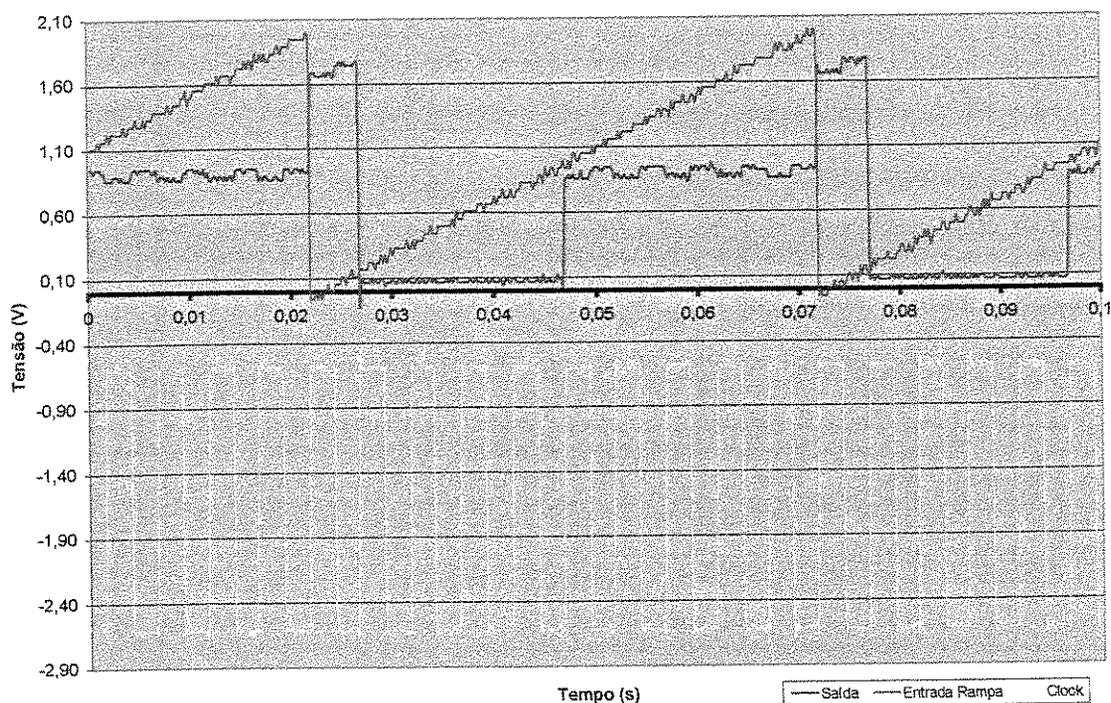


Figura 4.4

Como vemos na figura 4.4 o sinal de clock está deslocado da origem para uma melhor visualização do funcionamento da porta. Todas as medidas aqui mostradas foram efetuadas com um sinal de clock de 200Hz apenas por uma questão de padronização.

A frequência máxima do clock para operação do circuito ficou limitada em 12,1 kHz devido às capacitâncias de entrada das portas para qualquer sinal na entrada. Os sinais

entram diretamente no gate dos transistores P como podemos observar na figura do circuito Alfatopo (capítulo 1), e essas capacitâncias dos transistores não são descarregada gerando essa baixa frequência de operação.

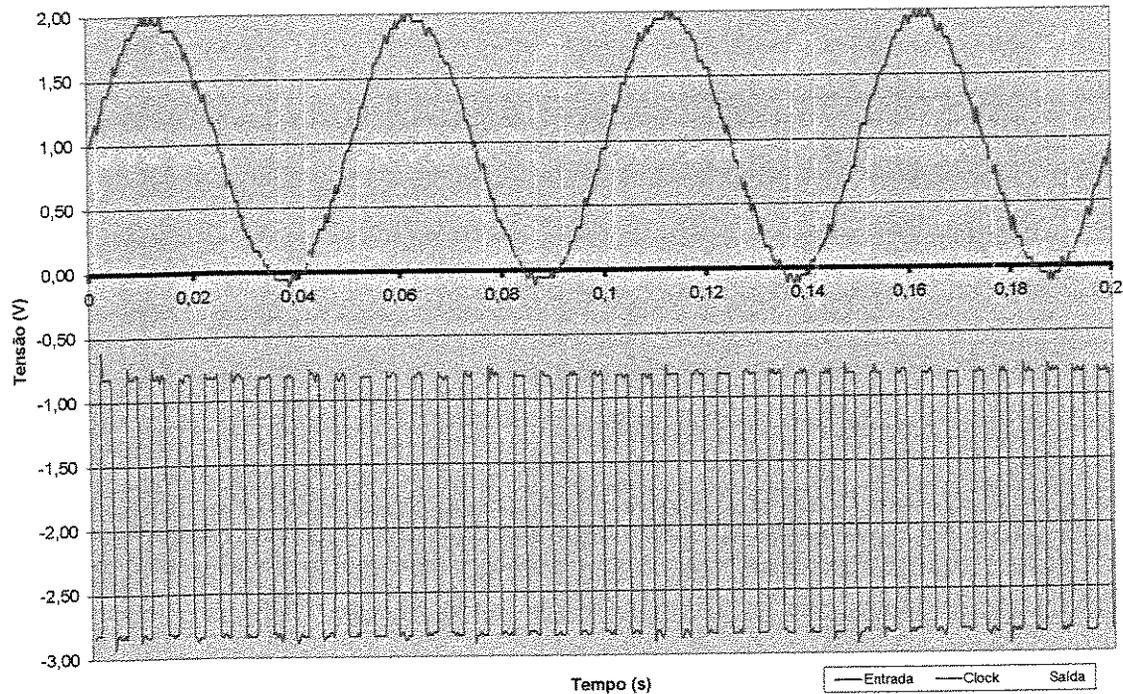


Figura 4.5

A figura 4.5 segue todas as orientações anteriores. Mudamos apenas o sinal de entrada e aqui observamos claramente a faixa na qual o circuito considera os valores lógicos. Quando a senoide passa pelo valor "0", vemos que a saída está no valor "2", mas por um tempo de dois sinais de clock porque a passagem da entrada pelo valor "0" é muito breve. Conforme essa entrada cresce em direção ao valor "1" vemos que esse limite na saída vai até a transição para o valor "0". Fica claro que o valor "2" tem uma faixa de valores muito grande em relação aos outros dois valores, basta observar que a saída fica muito mais tempo em valor "1" do que nos outros.

Em todas as figuras mostradas (independentemente do sinal de entrada, frequência e etc...) temos uma igual situação de polarização que é de:

- $V_{dd} = 2.484V$
- $V_{pol1} = 1.46V$
- $V_{pol2} = 1.2971$

Estes valores confirmam o que foi observado anteriormente para a porta Topo, mostrando que a porta Alfatopo também tem uma faixa de polarização com valores próximos dos da porta Topo.

4.3 Toggle Switch

Em nosso Integrado quando ligamos a saída com a entrada criamos um oscilador ternário (como em qualquer Flip-Flop tipo D), ou seja cada sinal de clock corresponderá a uma alteração na saída do circuito criando assim um sinal ternário com um clock de 1/3 do original

Na figura 4.6 podemos observar o circuito do Flip-Flop ligado em Toggle Switch oscilando sob uma frequência de 1 kHz. Utilizamos esta frequência para uma melhor visualização dos valores, mas este circuito atinge até 12,1 kHz. A frequência máxima de funcionamento do circuito ficou muito baixa devido à capacitância de entrada das portas básicas Topo e Alfatopo.

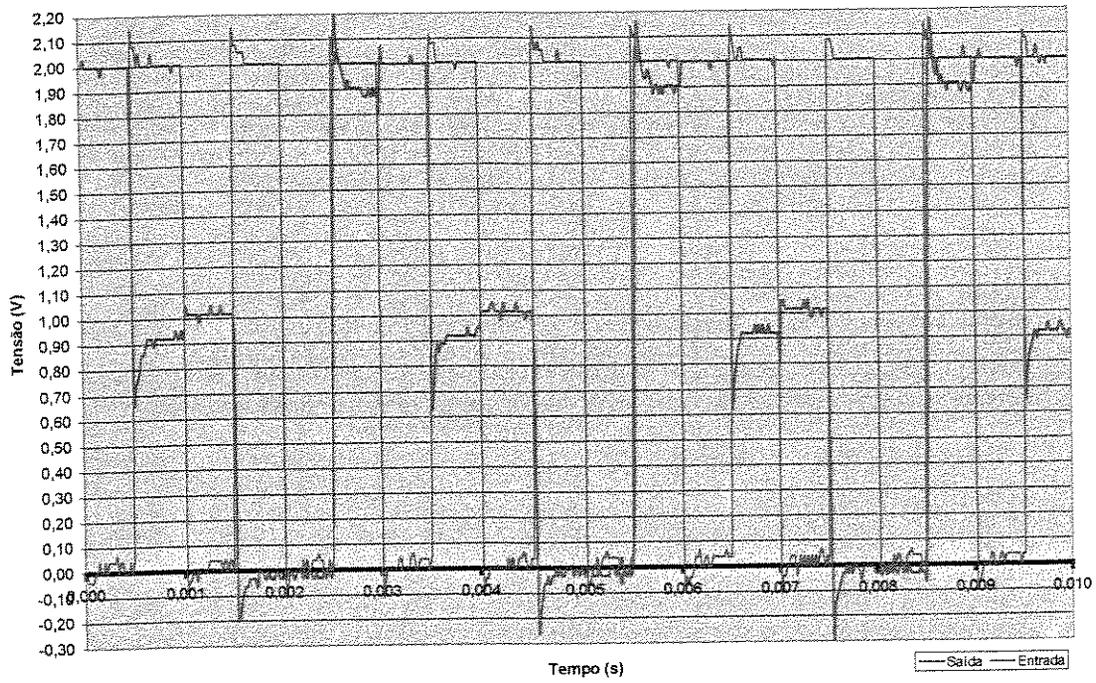


Figura 4.6

Na figura 4.7 podemos observar o mesmo circuito na frequência limite, acima desta o valor "2" desaparece e não podemos considerar este como um funcionamento adequado.

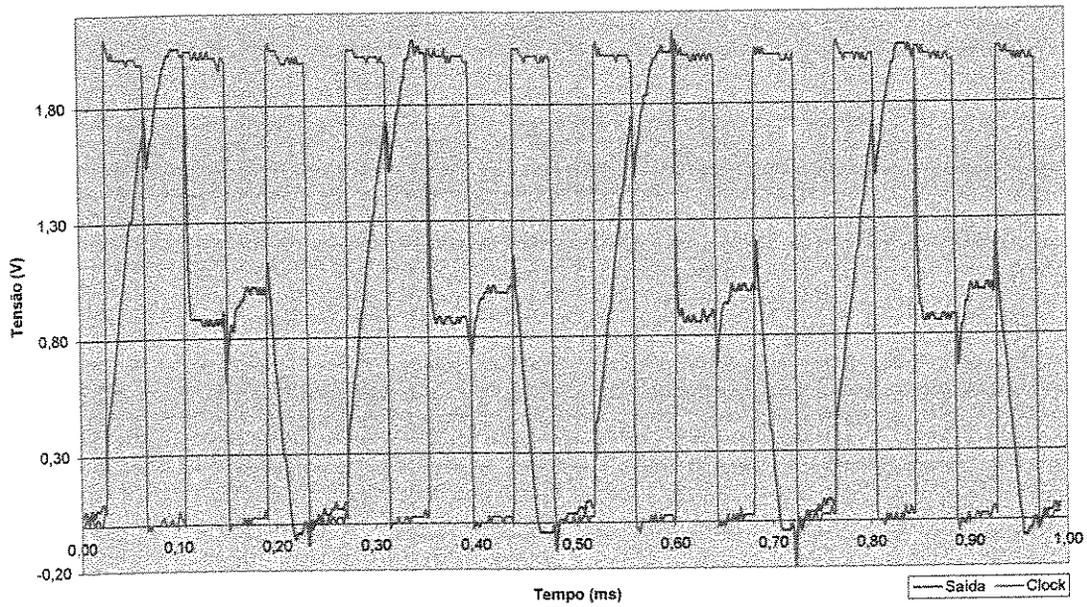


Figura 4.7

Procuramos medir os tempos de subida e descida do circuito, para uma frequência onde ainda poderíamos considerar como funcionamento normal, neste caso temos somente o Time Rise do valor "0" para o valor "2" porque o circuito oscila sempre com as mesmas formas das figuras 4.6 e 4.7.

Time Rise : "0" - "2" - 40us.

Time Fall : "2" - "1" 2us.

Time Fall : "1" - "0" 20us.

Temos que salientar que o Flip-Flop em Toggle Switch tem sua oscilação ocorrendo como está mostrado na figura 4.6, oscilando primeiro para valor "2" depois caindo para os dois outros valores, devido à escolha da realimentação da saída Base; caso nós tivéssemos realimentado com a saída Topo a saída seria invertida, ou seja oscilando primeiramente em valor "0" e depois mudando para os outros valores "1" e "2".

4.4 Somador.

Destacamos que o meio somador ternário não mostrou um comportamento adequado, ou seja, não foi possível ajustar as tensões de alimentação e polarização para que pudéssemos obter a operação de soma desejada.

Conforme vimos anteriormente cada porta básica possuía diferentes tensões de polarização; através dos testes pudemos quantificar as faixas de operação das portas Topo e Alfa3topo e também verificar que eram coincidentes. A terceira porta básica em nossos circuitos é a porta Alfa3topo; infelizmente não pudemos efetuar uma medida direta dos

valores das tensões, mas através dos testes pudemos concluir que seus valores são diferentes das outras portas, inviabilizando o circuito do Somador.

Não foi possível efetuar uma medida direta na porta Alfatopo por uma falha de Lay-Out, pois uma das entradas da porta não foi conectada com o seu respectivo Pad (faltou uma ligação entre os dois níveis de metal), ficando assim isolada dos sinais ternários necessários para o teste.

Conclusão

Neste trabalho desenvolvemos uma porta ternária universal, ou seja através dela pudemos construir todos os circuitos desejados dentro da lógica ternária, utilizando-se desta porta construímos um Flip-Flop e um Somador. Mostramos que esses circuitos em lógica multi-valores são viáveis em termos de implementação, necessitando apenas estudos para aumentar o seu desempenho com relação à frequência.

Por falhas de Lay-Out não foi possível mostrar os resultados do circuito Alfatopo, mas podemos supor que esse circuito tenha um desempenho adequado porque esta porta foi utilizada na construção do nosso Flip-Flop, sendo que este produziu um sinal satisfatório.

Podemos dizer que o fator que está limitando as aplicações possíveis do circuito do Flip-Flop é a necessidade de um clock binário com Rise time e Fall time específicos, como comentado no capítulo 2 dessa dissertação, impossibilitando a utilização desse circuito por exemplo em contadores ternários; quando ajustamos o sinal de saída do primeiro Flip-Flop para gerar um clock com $1/3$ da frequência para o próximo elemento as distorções que esses ajustes geram não são aceitas como um clock no segundo Flip-Flop, produzindo sinais distorcidos que não podem ser considerados como ternários.

Devido ao fato de termos colocados três Flip-Flops em cada C.I. pudemos observar diferenças significativas em suas tensões de polarização (V_{pol1} e V_{pol2}) para que esses pudessem funcionar de forma adequada. Essas diferenças nos sugerem que existem dispersões no processo de implantação iônica da Foundry.

Uma continuação natural deste trabalho seria a utilização dos transistores como elementos de chaves diminuindo assim o consumo de potência e possibilitando um grande salto na frequência dos elementos; por uma questão de comparação foi desenvolvido um

circuito Gama-Topo ternário seguindo essa filosofia [Yacoub], nas simulações o sinal desse circuito comportou-se de uma forma bastante satisfatória com um excelente desempenho com relação à frequência, temos apenas que ressaltar que foram necessários quarenta (40) transistores para desenvolver tal circuito, muito acima dos nove (9) transistores do nosso circuito.

Para um futuro trabalho podem-se desenvolver algumas melhorias, alterações nos circuitos que minimizem as capacitâncias das entradas, tentando assim descarregar os capacitores, para tentar aumentar a resposta em frequência do circuito. Temos aqui que salientar que as diferenças nas polarizações das portas Topo, Alfatopo e Alfa3topo não foram previstas por falha dos modelos da Foundry, há a necessidade da utilização de modelos mais completos que possam assim simular o real comportamento dos circuitos, evitando assim a falha ocorrida com a porta Alfa3topo desta dissertação.

Bibliografia.

- [Allen93] Allen, P. E., D. R. Holberg "CMOS Analog Circuit Design" – Holt, Rinehart and Winston, INC., pp: 219-226, 1993.
- [Biazon00] Biazon, A. J. Filho, "Multi-Valued Half Adder circuit using a CMOS Universal gate" - XV International Conference on Microelectronics and Packaging, Manaus, Brazil, pp. 125-129 - Set. 2000.
- [Epstein74] Epstein, G., Frieder, G., Rine, D.C. "The Development of Multiple-Valued Logic as Related to Computer Science" - Computer, Sept. 1974, pp: 20-32
- [Labview96] Manual do Labview - National Instruments, 1996.
- [Lukasiewicz20] Lukasiewicz, J., "O logice trójwartościowej", *Ruch Filozoficzny*, english translation: On three valued-logic, in L. Borkowski (ed), *Select Works*, North-Holland, Amsterdam, Vol. 5, pp. 169-171, 1920.
- [Martins99] Jorge, A. Martins, Comunicação Interna, Unicamp/Feec/Demic, Julho 1999
- [Martins00] Jorge, A. Martins, "A Universal CMOS gate for Multi-Valued logic (MVL) circuits" - XV International Conference on Microelectronics and Packaging, Manaus, Brazil, pp. 119-124, Set. 2000.
- [Nascimento00] Luciana Prado do Nascimento, Comunicação Interna, Unicamp/Feec/Demic, Agosto 2000
- [National I.99] "Measurement and Automation Catalogue" - National Instruments, 1999.
- [Post20] Post, E. L., "Introduction to a general theory of elementary propositions" - American Journal of Mathematics, Vol. 43, pp. 163-185, 1920.

- [Serran96] Serran, N. V., "Circuitos Digitais Ternários baseados na Álgebra de Post, estudo, definição de operadores e implementação" - Tese de Doutorado, Universidade de Campinas, SP, Brasil, 1996
- [Serran97] Serran, N. V., A. Martins Jorge and J. A. Siqueira Dias, "A Proposal for the Implementation of Ternary Digital circuits" - *Microelectronics Journal*, Vol. 28, N° 5, pp. 533-541, 1997.
- [Spice95] Rashid, M. H. "Spice for circuits and electronics using Pspice" - Englewood Cliffs. Prentice Hall, 1995.
- [Tanner96] Manual Tanner - Tanner Research, Inc., Version 6, 1996.
- [Taub82] Taub, H., D. Schilling, "Eletrônica Digital" - Ed. MacGraw Hill, 1982.
- [Thoidis98] Thoidis, I., D. Soudris, I. Karafyllidis, S. Christoforidis and A. Thanailakis, "Quaternary voltage-mode CMOS circuits for multiple-valued logic" - *IEE Proceedings – Circuits Devices Systems*, Vol. 145, N° 2, pp. 71-77, 1998.
- [Tocci98] Tocci, R. J., N. S. Widmer "Sistemas Digitais, Princípios e Aplicações" - Editora L.T.C. 1998
- [Yacoub00] Yacoub, M. N. R. D., "Proposta de Implementação de uma Lógica Ternária em tecnologia CMOS" - Tese de Doutorado, Universidade de Campinas, S.P., Brasil, 2000.

Anexo A

arbitrária.vi

O Gerador de Funções Arbitrárias permite a criação de suas próprias formas de onda nas duas saídas D/A da placa DAC - PC1200, chamados de canais 0 e 1. Com esse instrumento pode-se criar as formas de onda desejadas ou carregá-las de um arquivo já gravado. Para a opção de criação pode-se criar formas de onda através de uma subrotina que tem formas pré-gravadas (Senoidal, Quadrada, Rampa e Triangular) controlando também a amplitude, offset, número de períodos, defasamento e o duty cycle (para formas de onda Quadradas). Alternativamente pode-se criar manualmente a forma de onda desejada através de uma segunda subrotina. Uma vez criadas as formas de onda pode-se editar, salvar o arquivo ou carregar um arquivo previamente criado.

Uma vez criada a forma de onda, escolhe-se uma taxa de atualização, e poderá ter o sinal uma vez ou continuamente.

Consultando o Help do Labview pode-se aprofundar na utilização do Instrumento Virtual.

Painel Frontal

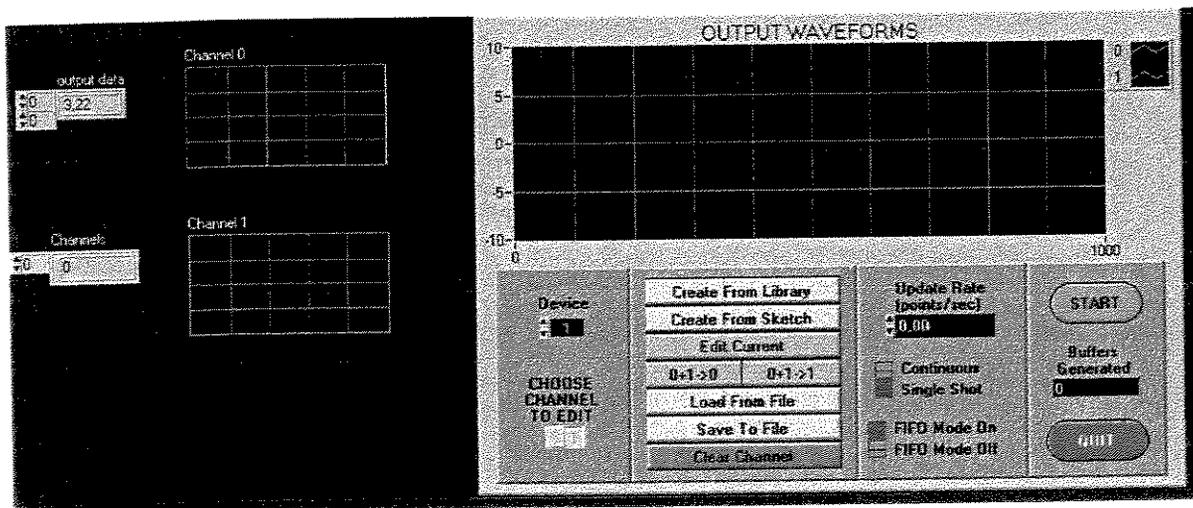
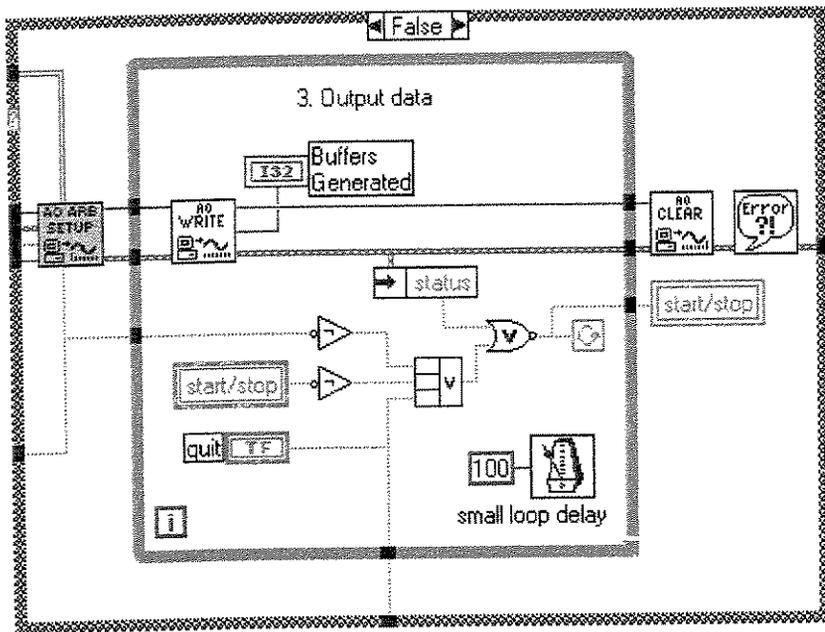
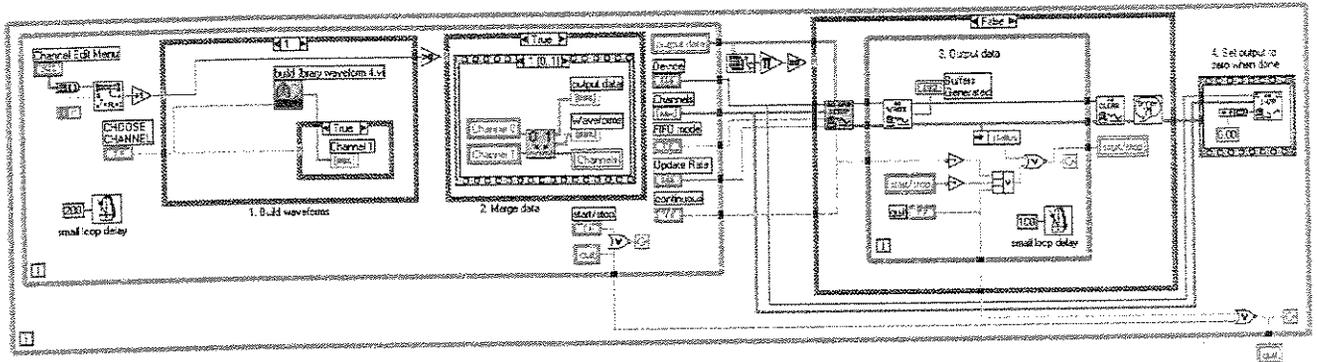
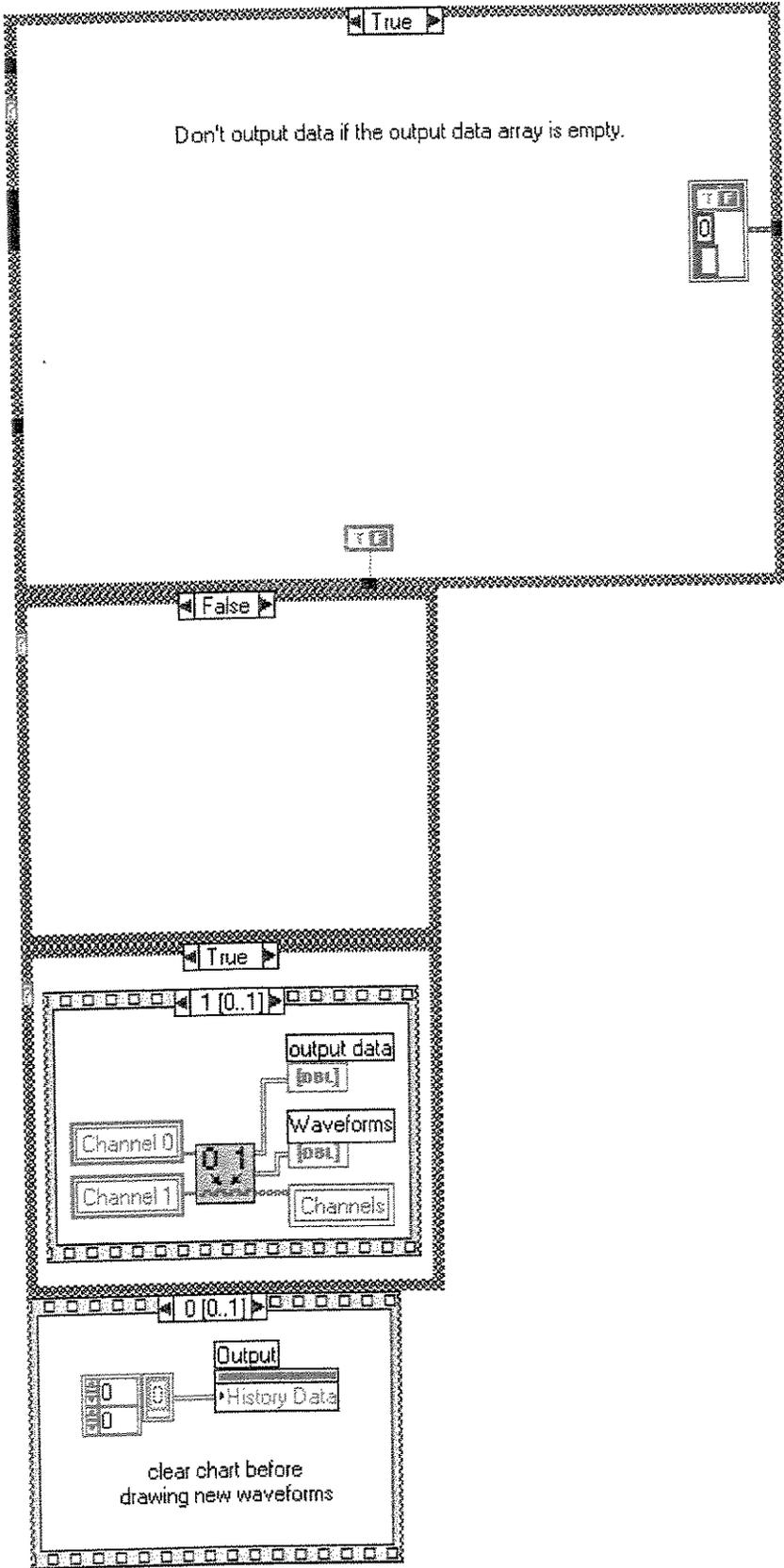
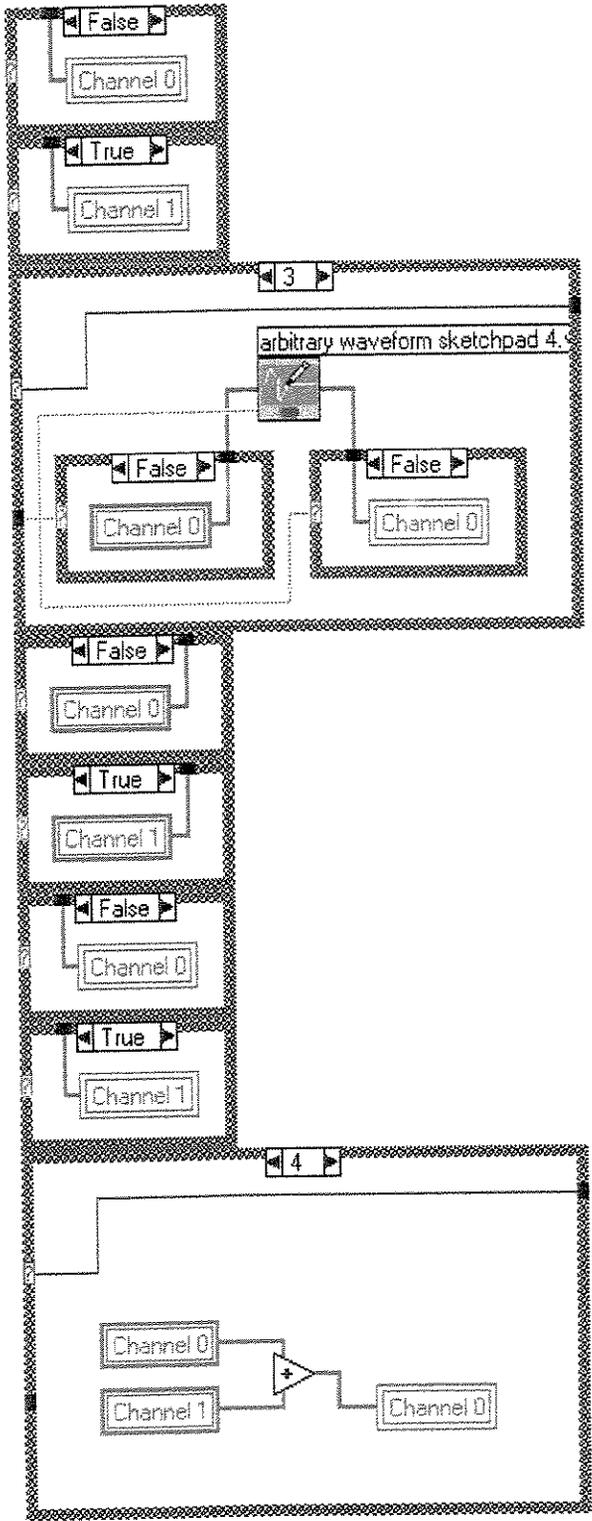
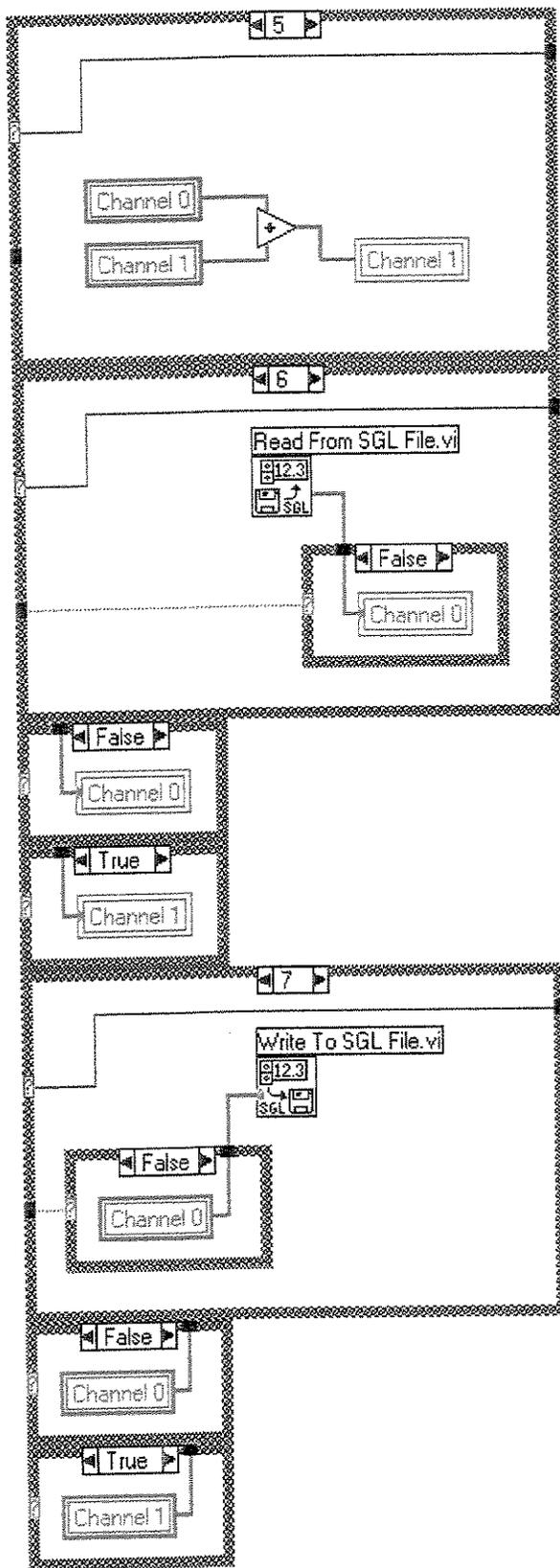


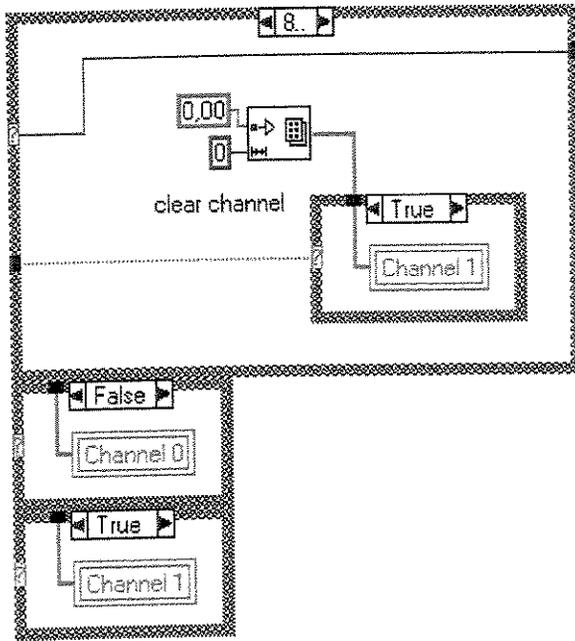
Diagrama de Blocos











Geratudo.vi

Este gerador permite a criação de formas de onda que complementam a V.I. anterior porque pode gerar ondas ternárias (além de todas as ondas anteriormente descritas), esses arquivos podem ser carregados na V.I. *arbitraria.vi* e assim carregadas nas duas saídas D/A da placa DAC - PC1200 chamados de canais 0 e 1.

Painel Frontal

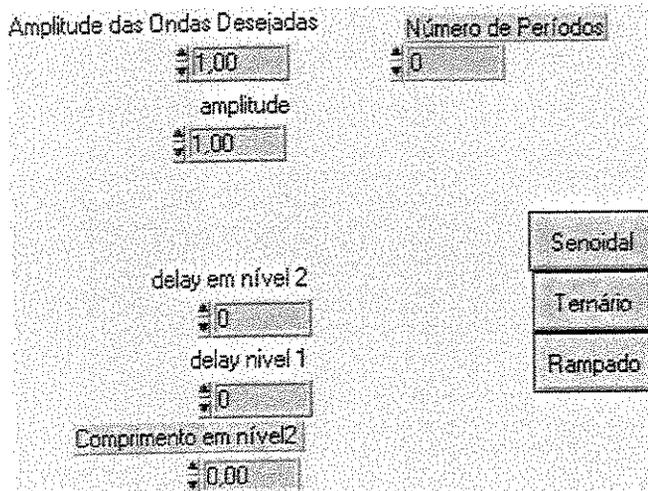
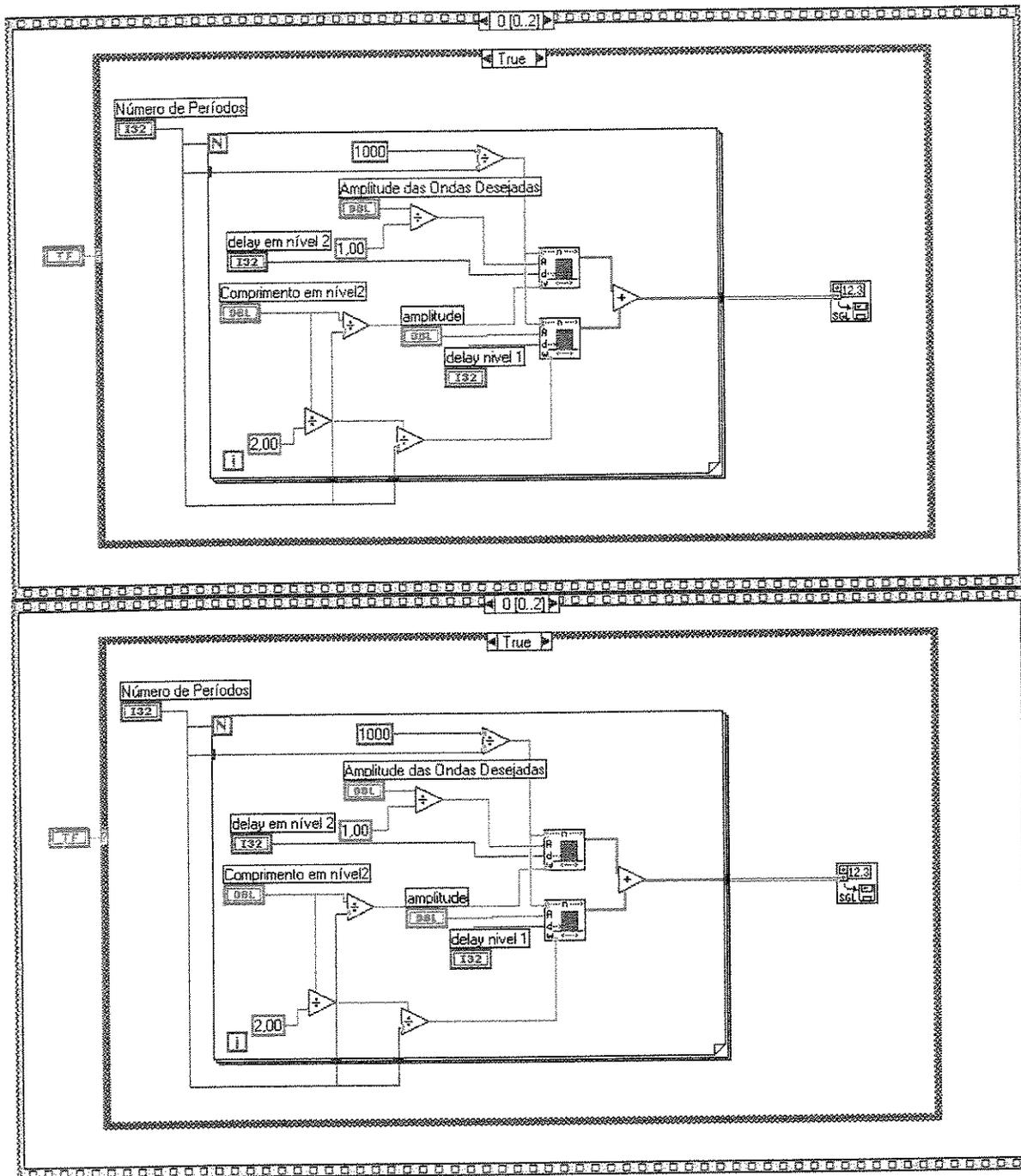
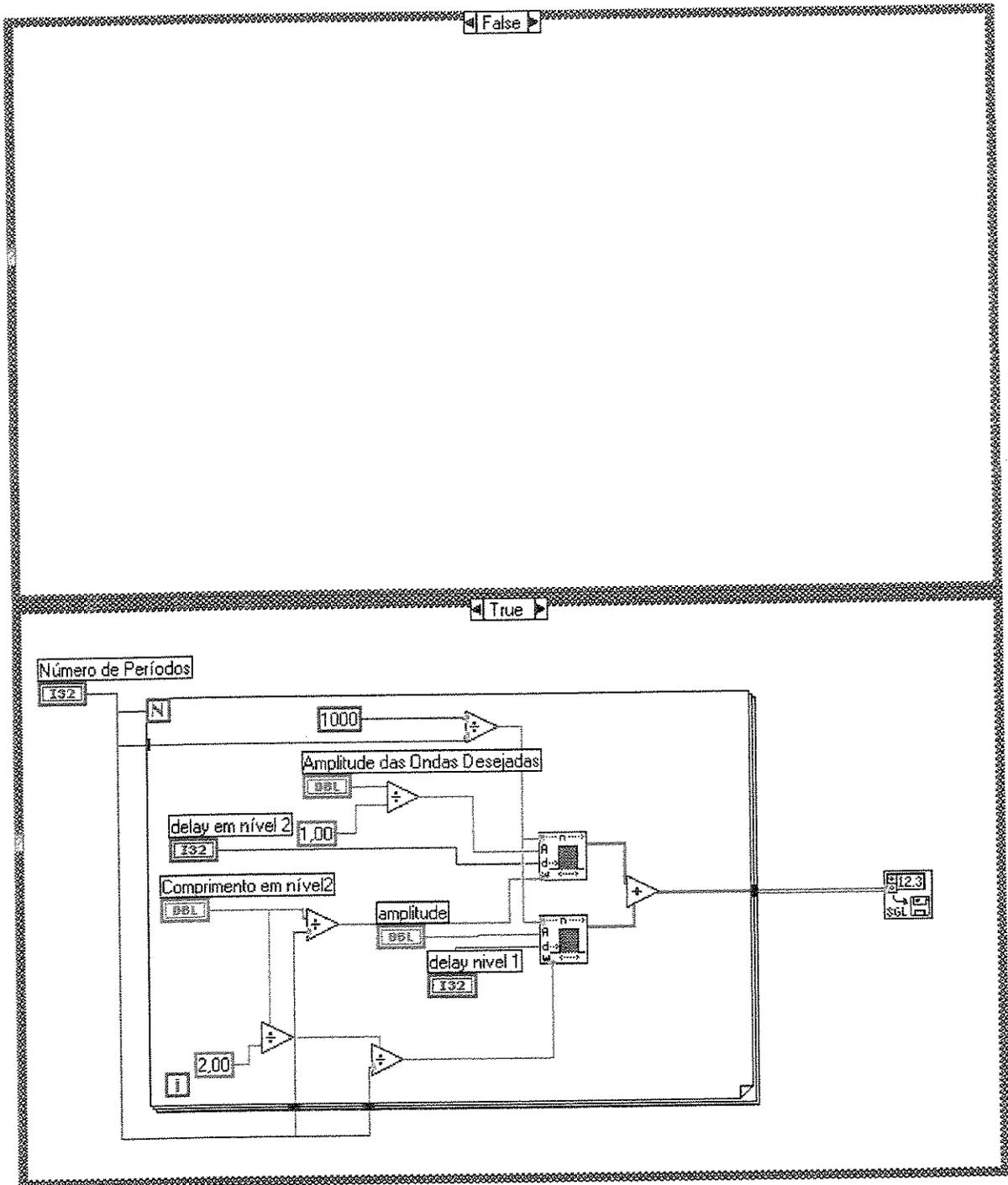
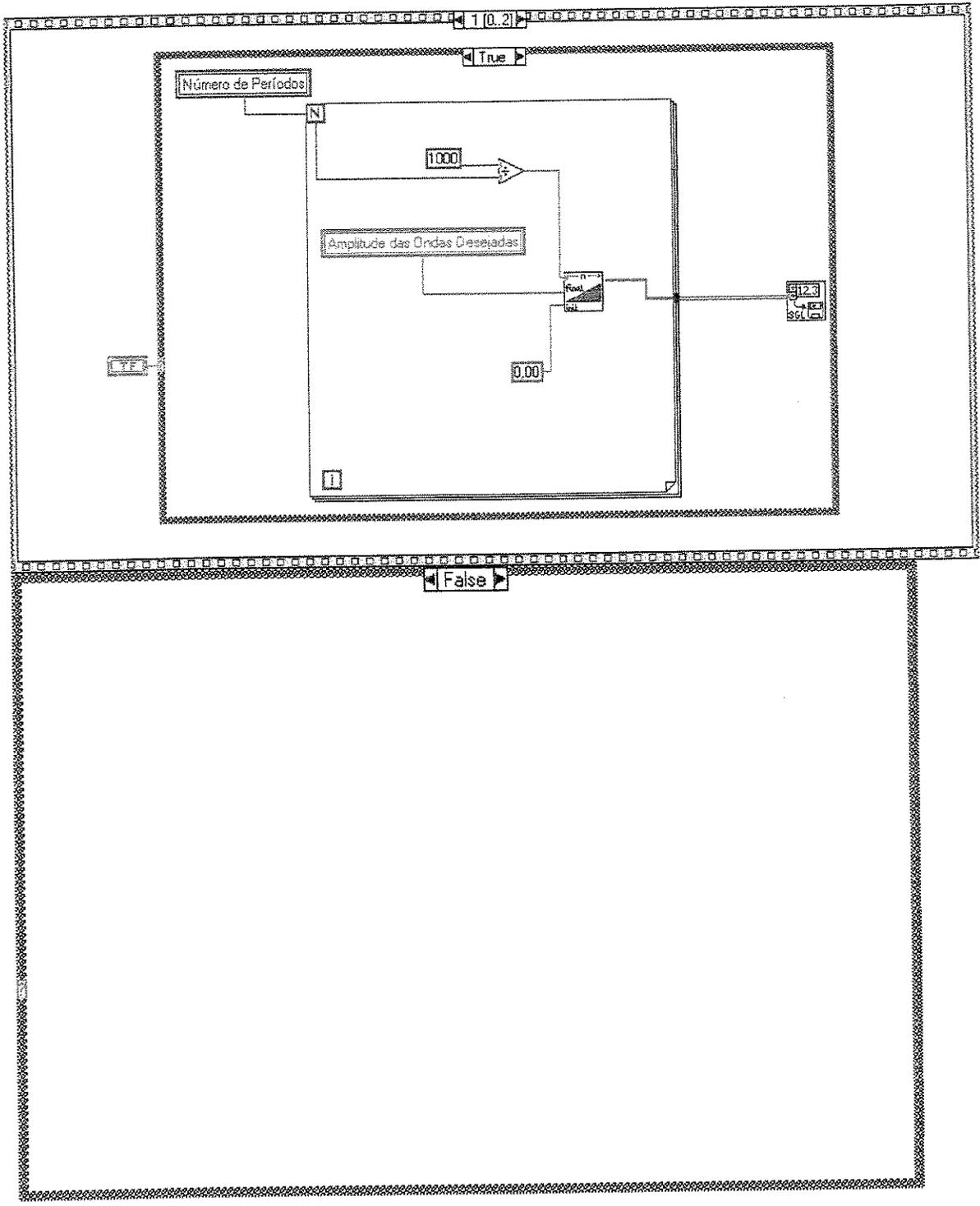
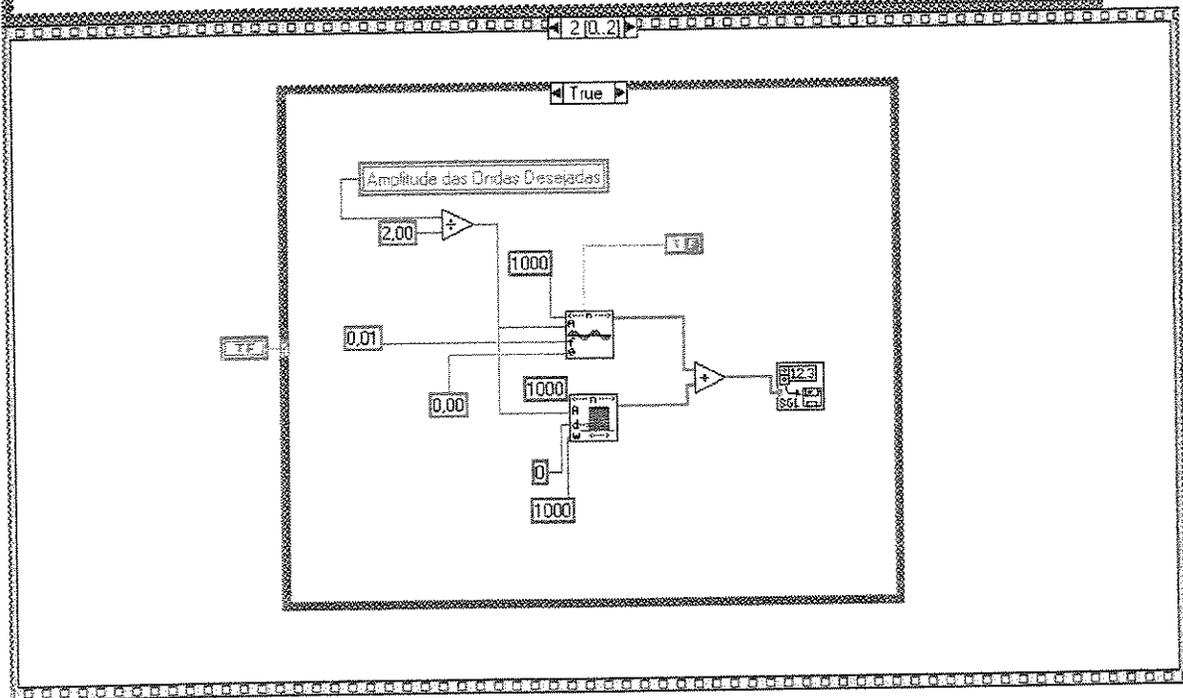
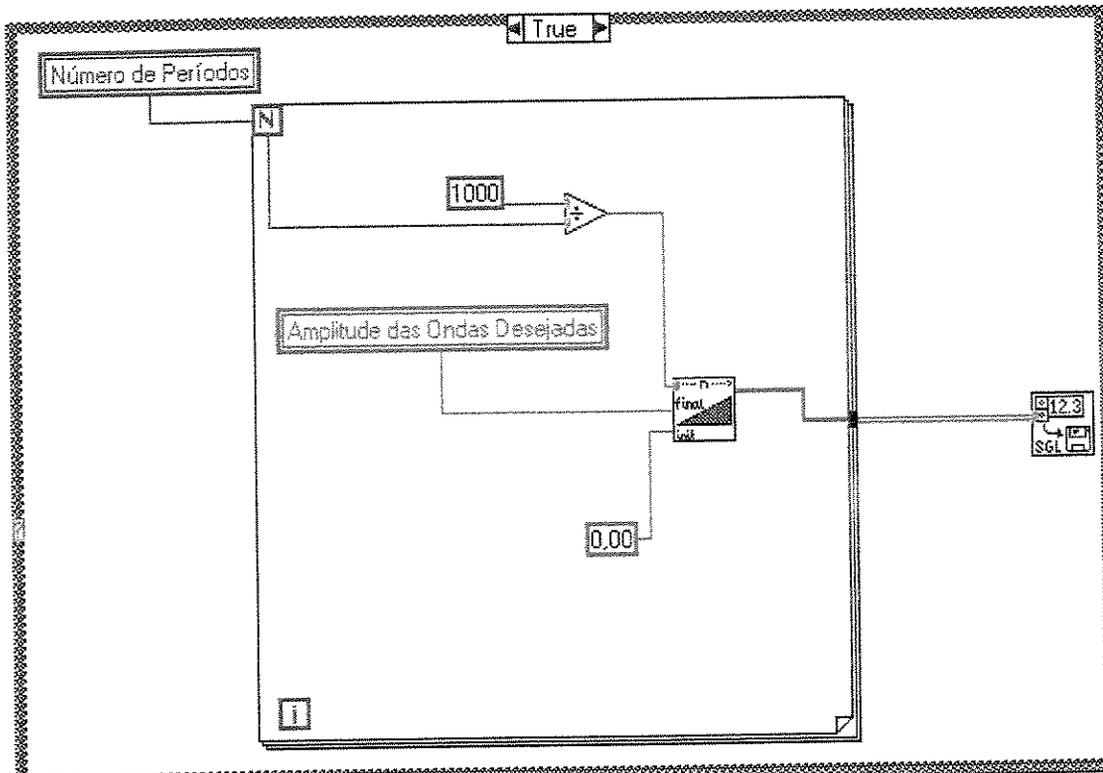


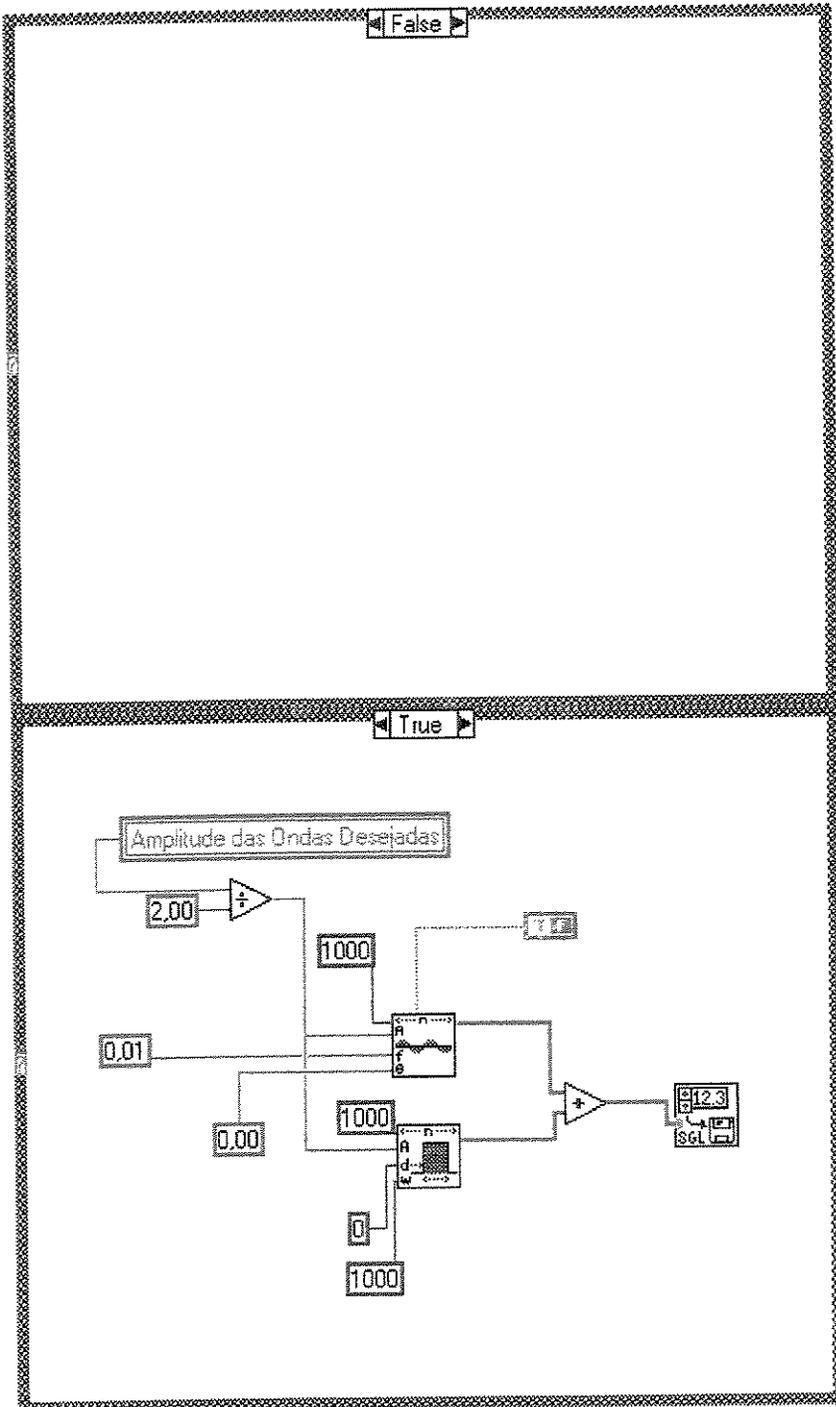
Diagrama de Blocos











medidas.vi

A função dessa V.I. é ler os dados dos canais escolhidos do osciloscópio. Existe a opção de se gravar as formas de ondas do osciloscópio, para a confecção da documentação desejada. Este arquivo gerado pode ser aberto em Excel, onde pode-se trabalhar com liberdade com esses dados.

Painel Frontal

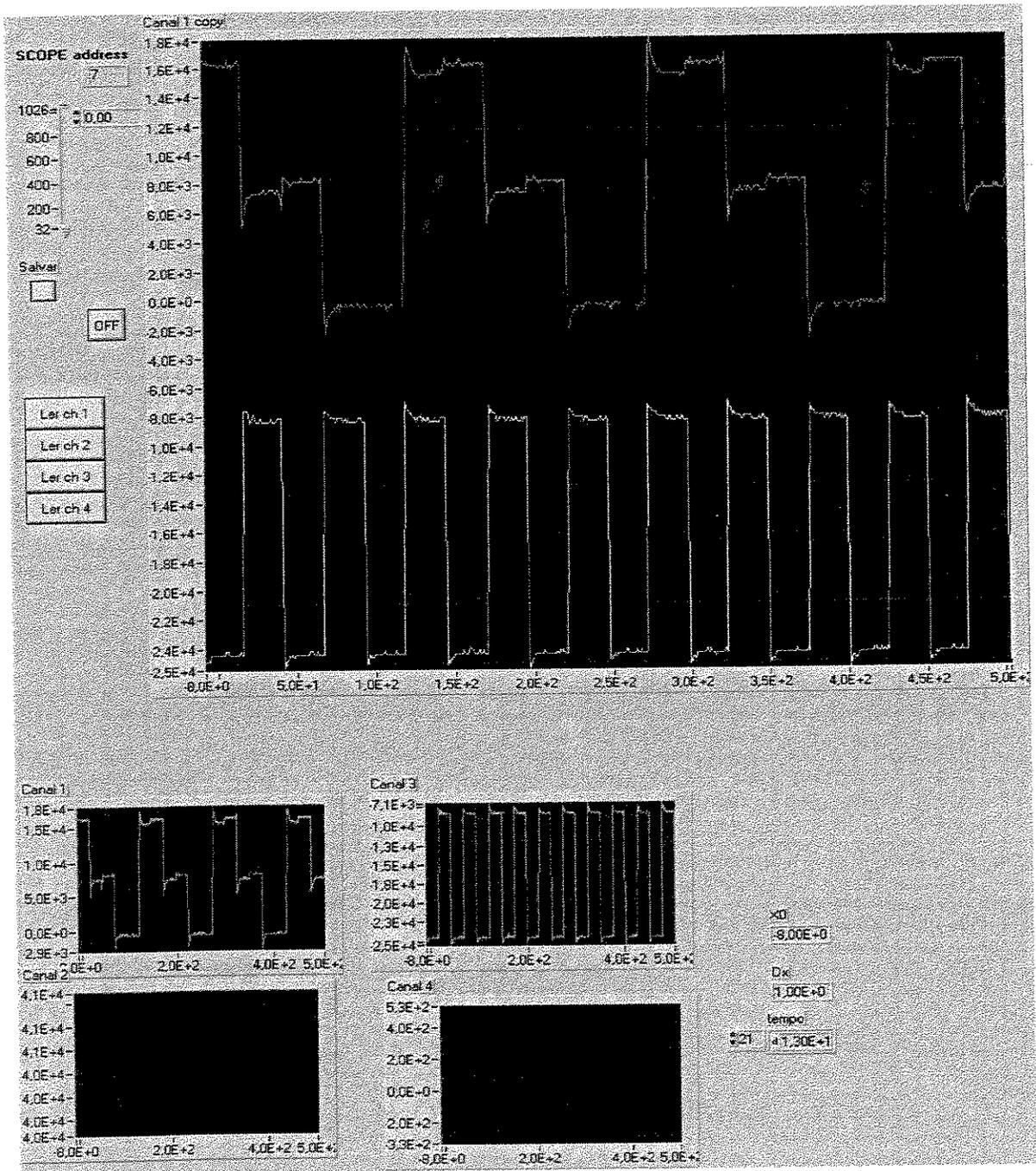
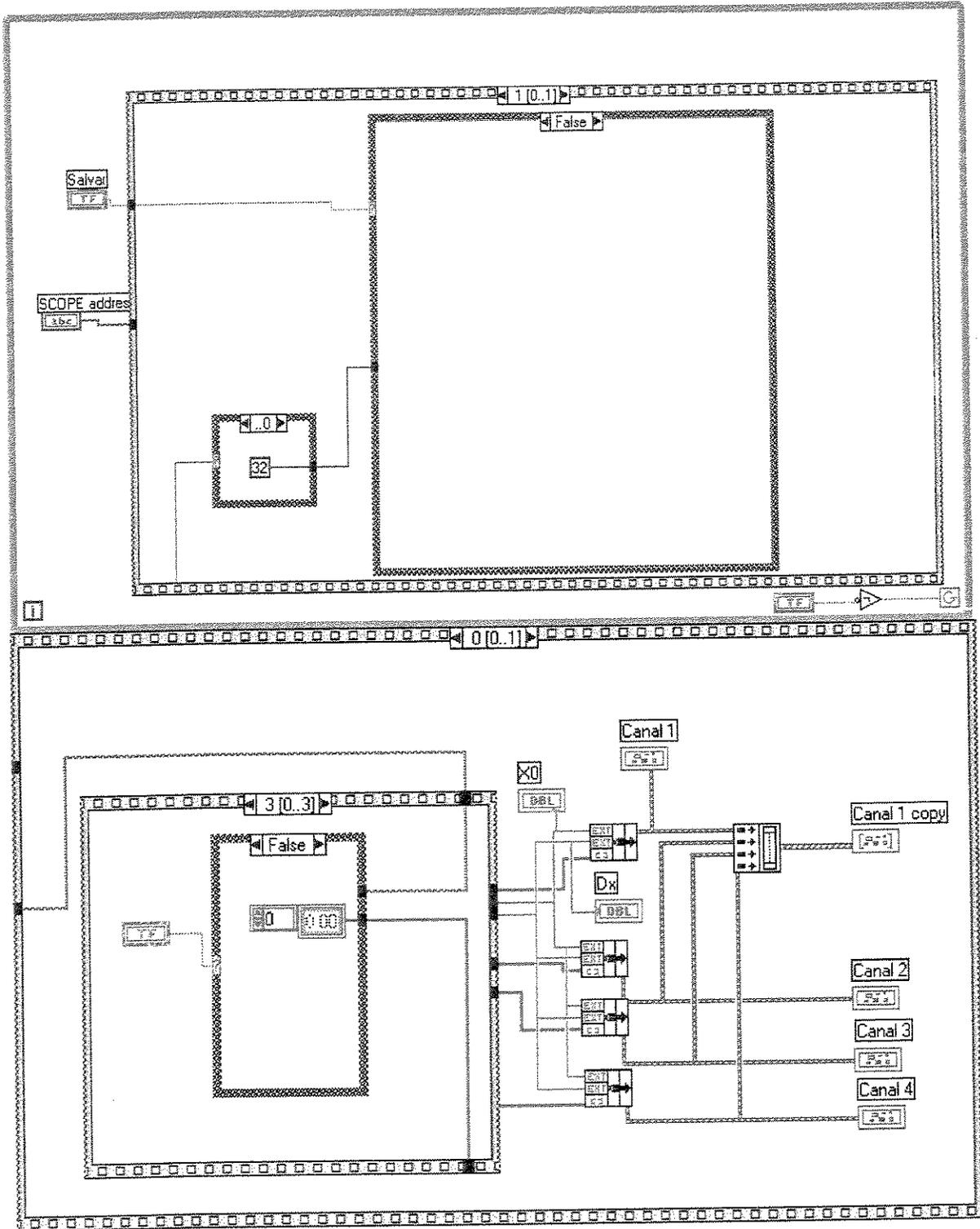
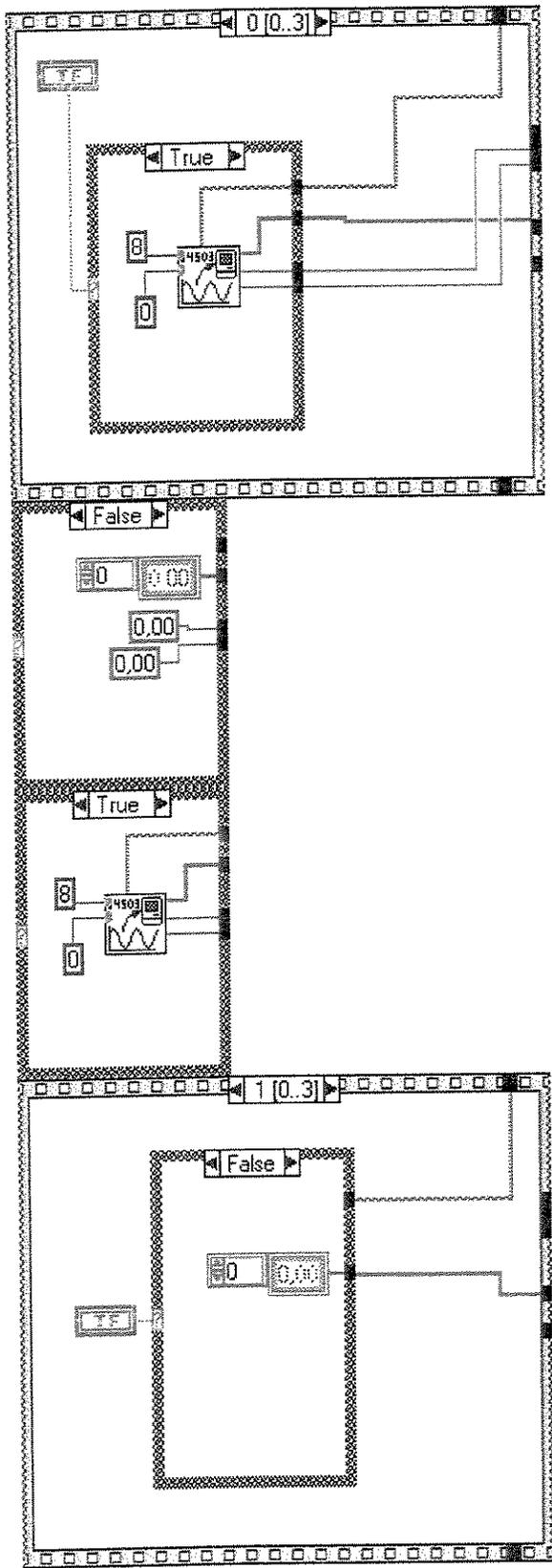
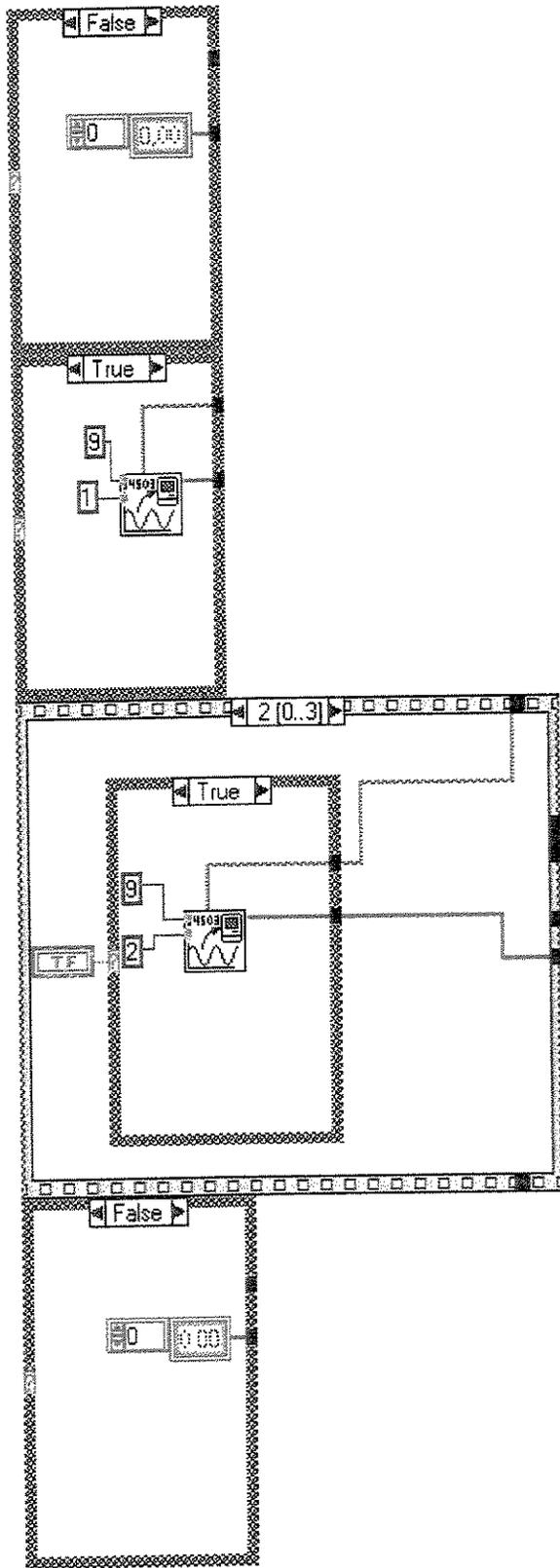
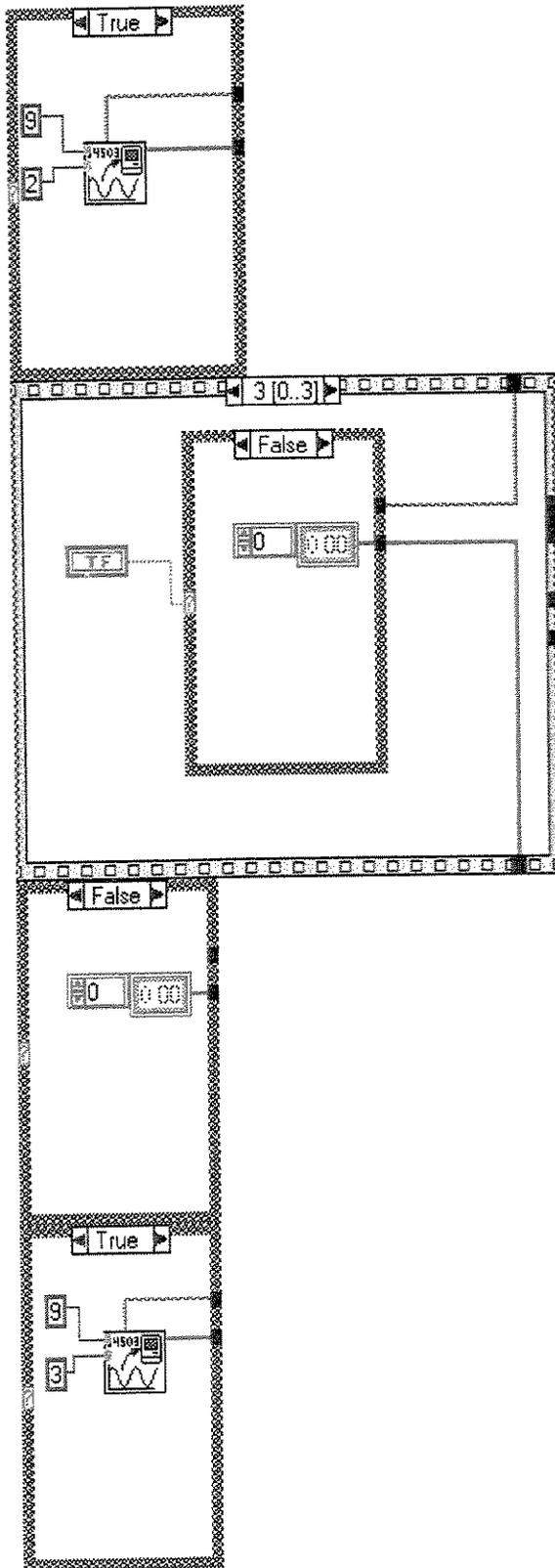


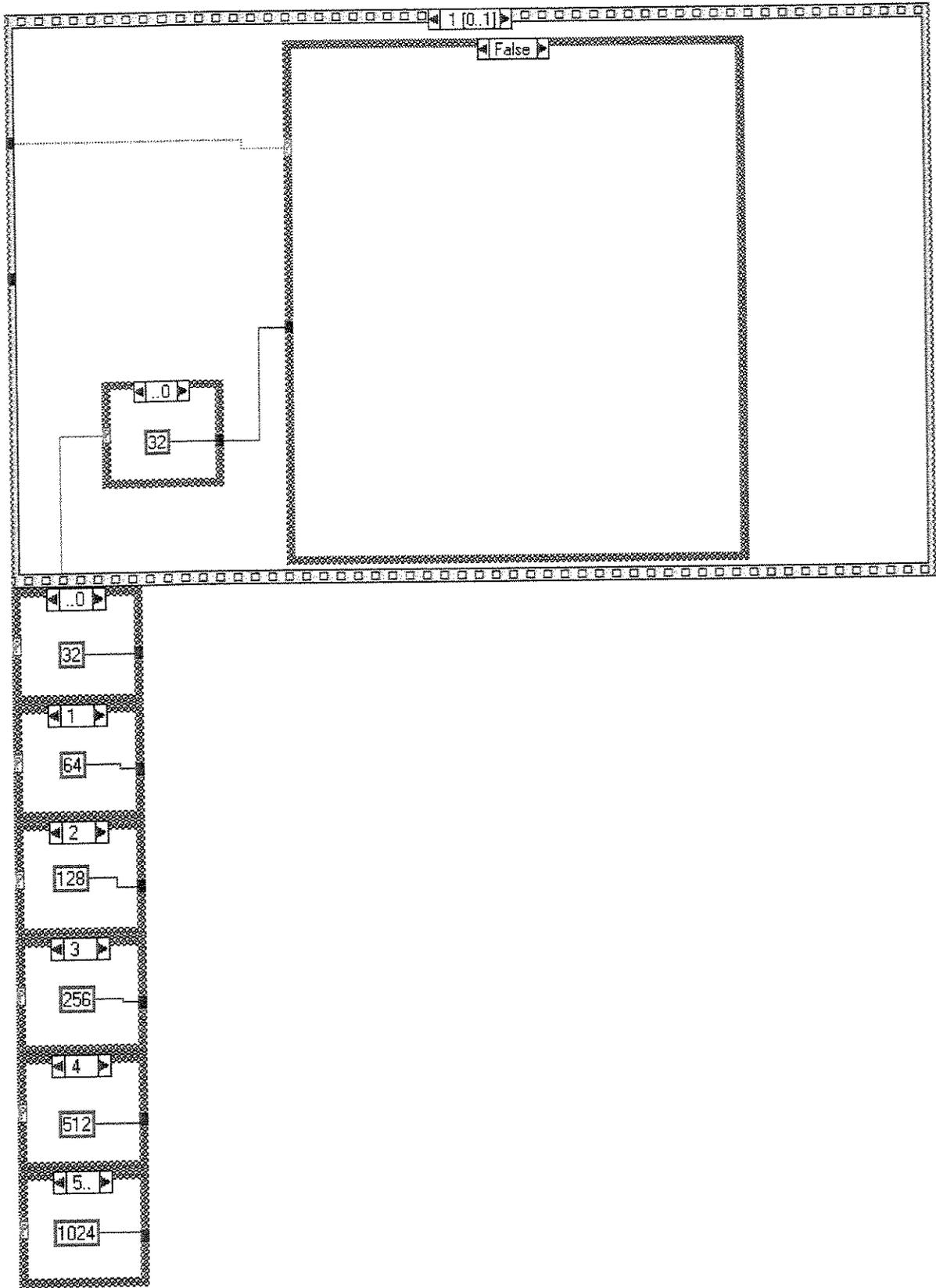
Diagrama de Blocos

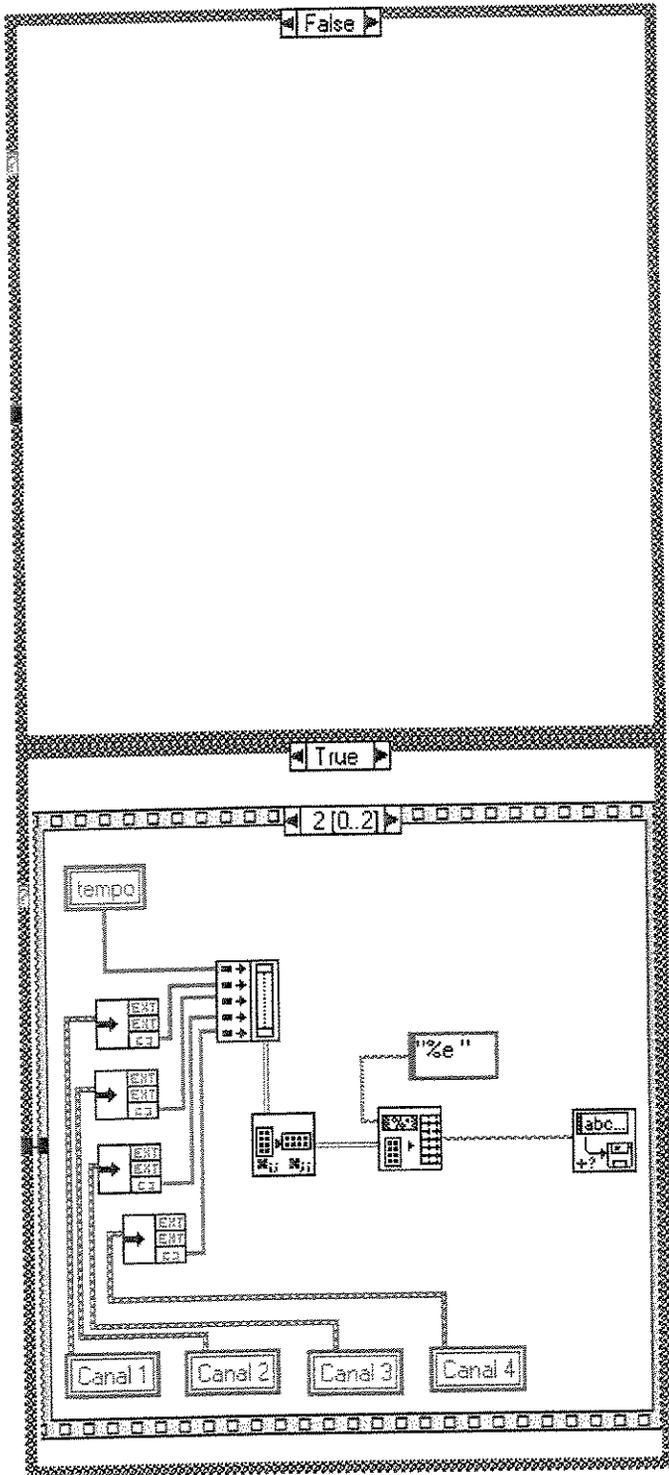


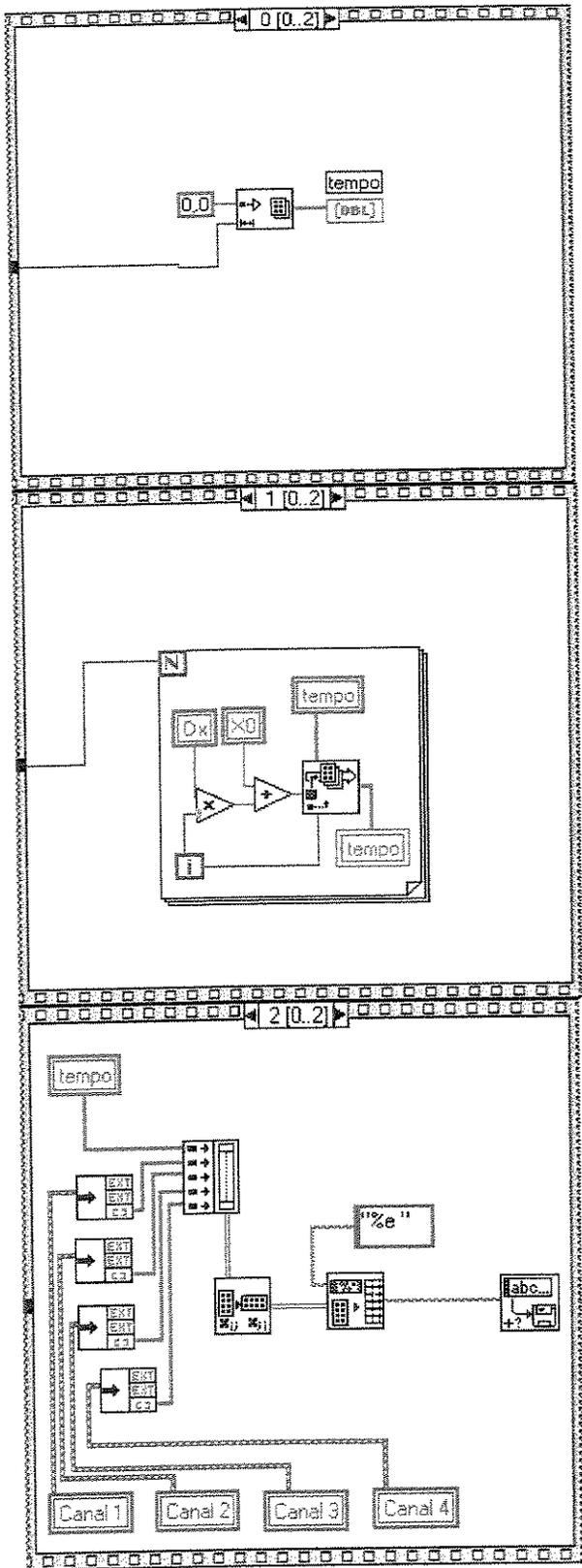












Anexo B

Curvas dos Transistores.

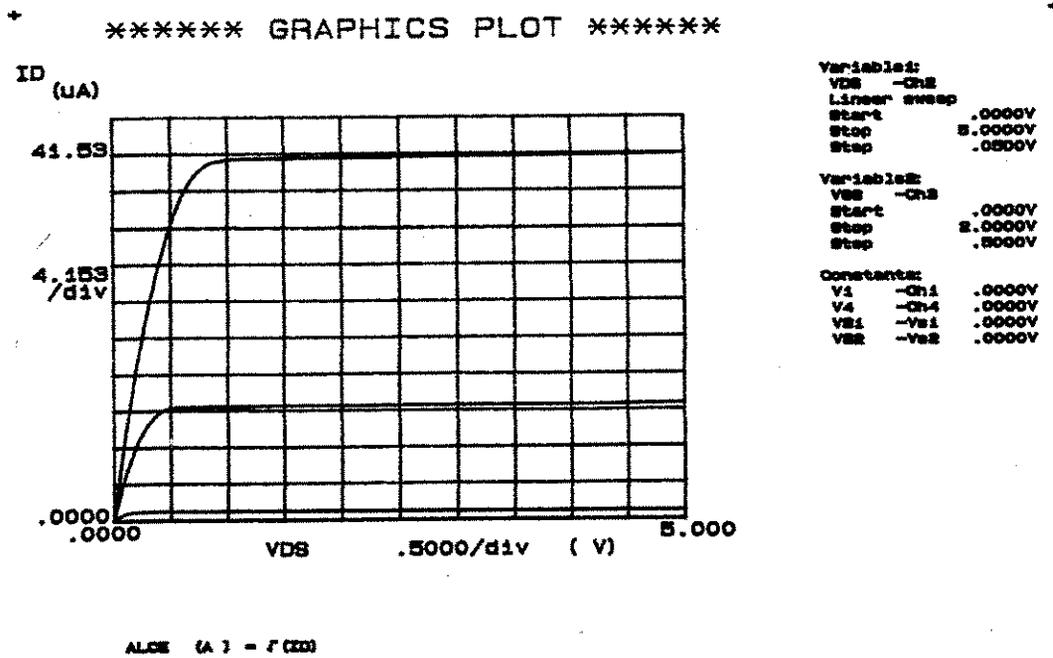


Figura 1: Id versus Vds do transistor N

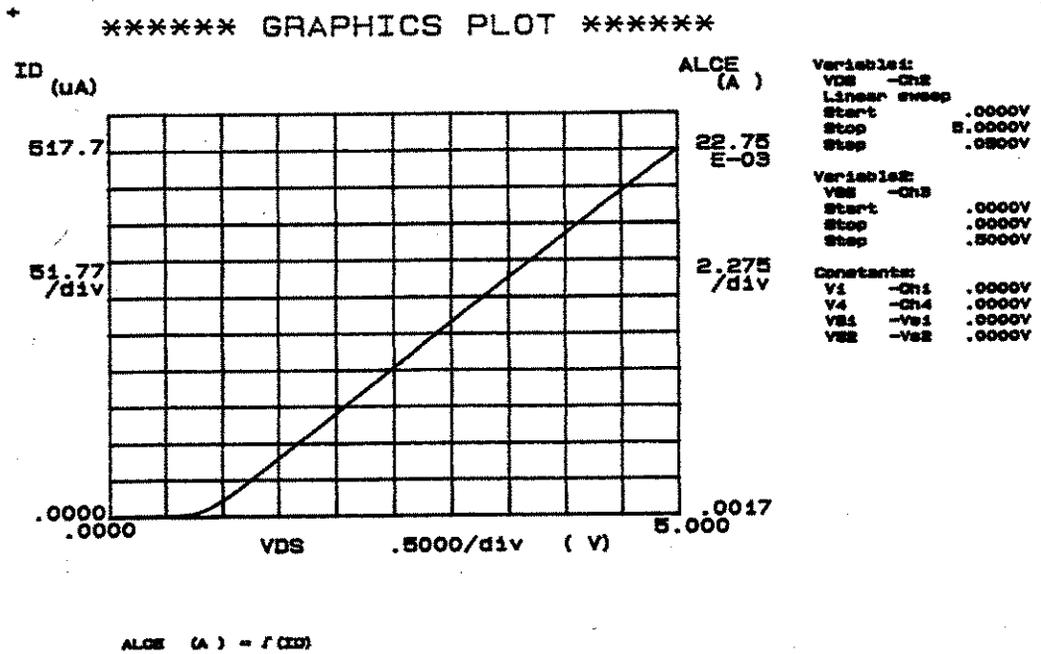


Figura 2: $\sqrt{I_d}$ versus V_{gs} do transistor N – Determinação da Tensão de Limiar

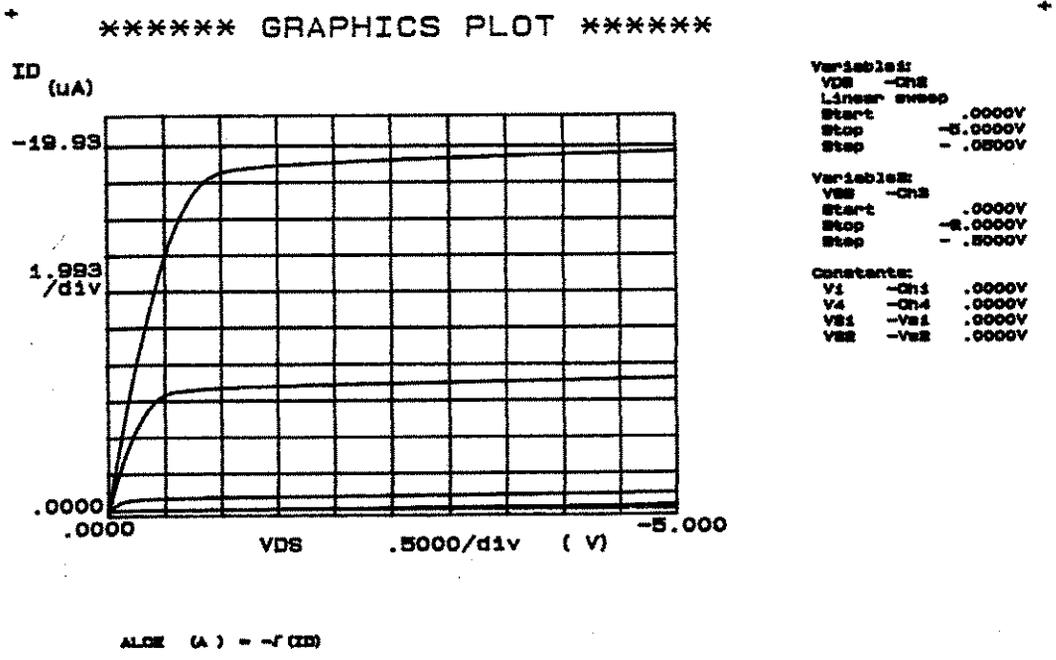


Figura 3: Id versus Vds do transistor P

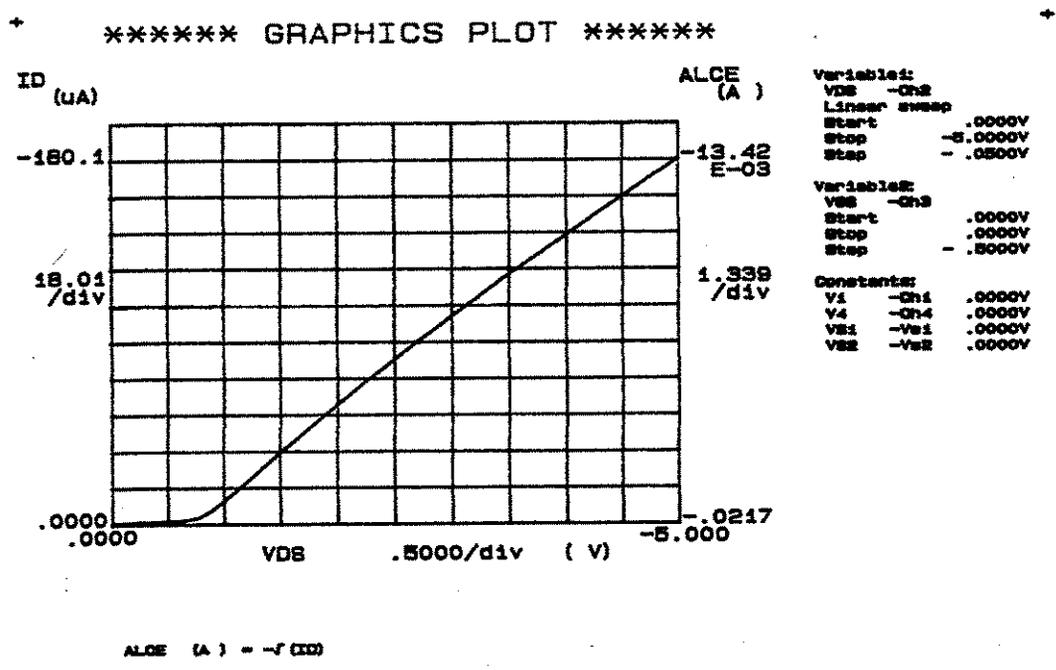


Figura 4: $\sqrt{I_d}$ versus V_{gs} do transistor P – Determinação da Tensão de Limiar

Lay-Out final e Foto do Integrado

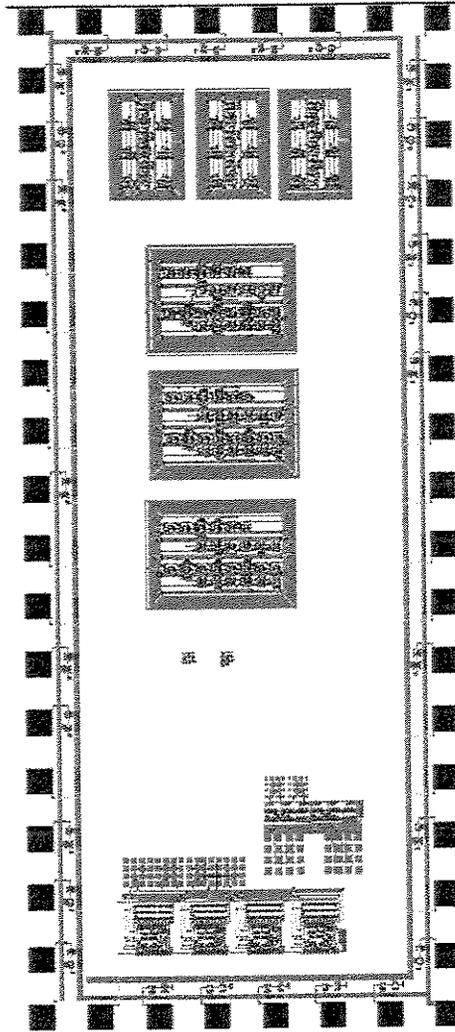


Figura 5: Lay-Out do Circuito Integrado

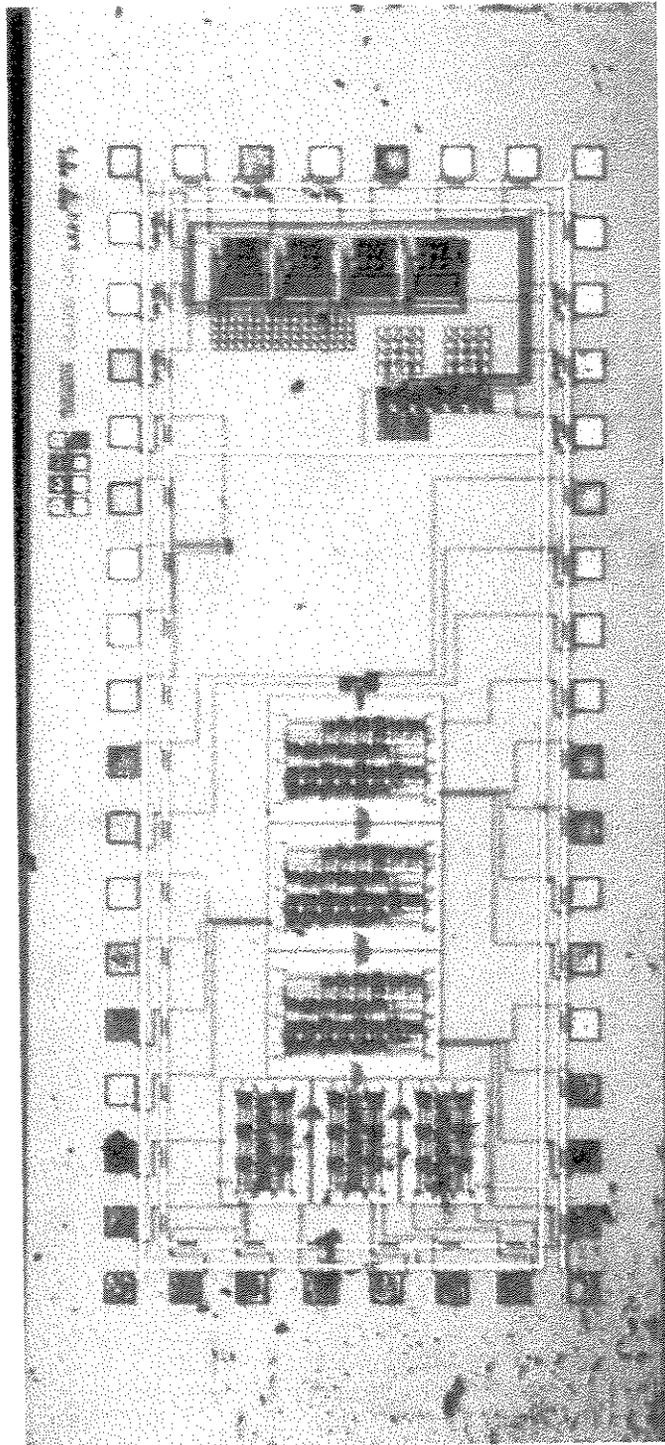


Figura 6: Fotografia do Circuito Integrado.