



UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
CURSO DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

Vinícius Nolêto de Araújo e Alex Silva Costa

LABORATÓRIO DE CIRCUITOS

BOA VISTA, RR
2022

Vinícius Nolêto de Araújo e Alex Silva Costa

LABORATÓRIO DE CIRCUITOS

Avaliação de Barramento de Circuitos Digitais, apresentado como requisito de obtenção de nota parcial da disciplina de Arquitetura e Organização de Computadores - DCC 301.

Orientador: Prof. Dr. Hebert Oliveira Rocha

BOA VISTA, RR

2022

LISTA DE ILUSTRAÇÕES

Figura 1 -	<i>Flip-Flop JK</i>	7
Figura 2 -	<i>Flip-Flop D</i>	7
Figura 3 -	Multiplexador.....	9
Figura 4 -	Porta Lógica <i>Xor</i>	10
Figura 5 -	Circuito Completo Somador 8 bits.....	11
Figura 6 -	Valores de Entrada Somador 8 bits.....	11
Figura 7 -	Secção de Circuito Somador 8 bits.....	12
Figura 8 -	Circuito Completo Memória ROM.....	13
Figura 9 -	Entradas e Saídas Memória ROM.....	13
Figura 10 -	Valor de Memória.....	14
Figura 11 -	Memória RAM.....	15
Figura 12 -	Entradas e Saídas Memória RAM.....	15
Figura 13 -	Banco de Registradores.....	16
Figura 14 -	Circuito Completo Somador 8 bits.....	17
Figura 15 -	Valores de Entrada Somador 8 bits.....	17
Figura 16 -	Unidade de Controle 16 bits.....	18
Figura 17 -	Unidade de Controle 16 bits Classe R Caso 1.....	19
Figura 18 -	Unidade de Controle 16 bits Classe R Caso 2.....	19
Figura 19 -	Unidade de Controle 16 bits Classe R Caso 3.....	19
Figura 20 -	Unidade de Controle 16 bits Classe R Caso 4.....	20

Figura 21 - ULA 8 bits.....	20
Figura 22 - ULA 1 bit.....	21
Figura 23 - Extensor de Sinal.....	22
Figura 24 - Máquina de Estados Herbert.....	23
Figura 25 - Conceito Máquina de Estados Herbert.....	23
Figura 26 - Máquina de Estados Herbert (Caso 1).....	24
Figura 27 - Máquina de Estados Herbert (Caso 2).....	24
Figura 28 - Máquina de Estados Herbert (Caso 3).....	24
Figura 29 - Máquina de Estados Herbert (Caso 4).....	24
Figura 30 - Máquina de Estados Herbert (Caso 5).....	24
Figura 31 - Contador Síncrono.....	26
Figura 32 - <i>Flip-Flop T</i>	27

LISTA DE TABELAS

Tabela 1 -	Tabela-Verdade Flip-Flop RS Completa.....	8
Tabela 2 -	Tabela-Verdade Flip-Flop JK Completa.....	8
Tabela 3 -	Tabela-Verdade Flip-Flop JK Simplificada.....	8
Tabela 4 -	Tabela-Verdade Flip-Flop D Completa.....	8
Tabela 5 -	Tabela-Verdade Flip-Flop D Simplificada.....	9
Tabela 6 -	Tabela-Verdade Multiplexador.....	9
Tabela 7 -	Tabela-Verdade XOR.....	10
Tabela 8 -	Tabela-Verdade Secção de Circuito Somador 8 bits.....	12
Tabela 9 -	Códigos ULA.....	22
Tabela 10 -	Tabela-Verdade Máquina de Estados Herbert.....	26

SUMÁRIO

SUMÁRIO	5
1. <u>INTRODUÇÃO</u>	6
2. <u>COMPONENTES</u>	7
2.1 <u>REGISTRADORES FLIP FLOP D E JK</u>	7
2.2 <u>MULTIPLEXADOR COM 4 OPÇÕES DE ENTRADA</u>	9
2.3 <u>PORTA LÓGICA DO XOR</u>	10
2.4 <u>SOMADOR 8 BITS COM VALOR 4</u>	11
2.5 <u>MEMÓRIA ROM</u>	13
2.6 <u>MEMÓRIA RAM</u>	15
2.7 <u>BANCO DE REGISTRADORES</u>	16
2.8 <u>SOMADOR 8 BITS</u>	17
2.9 <u>UNIDADE DE CONTROLE 16 BITS</u>	18
2.10 <u>ULA 8 BITS</u>	20
2.11 <u>EXTENSOR DE SINAL</u>	22
2.12 <u>MÁQUINA DE ESTADOS HERBERT</u>	23
2.13 <u>CONTADOR SÍNCRONO</u>	26
3. <u>CONSIDERAÇÕES FINAIS</u>	29
4. <u>REFERÊNCIAS</u>	30

1. INTRODUÇÃO

O presente relatório é referente aos circuitos desenvolvidos para a Avaliação de Barramento da disciplina Arquitetura e Organização de Computadores DCC 301. Os testes e descrições das lógicas binárias e aritméticas dos circuitos estão dispostos de acordo com as questões da avaliação, por isto não há necessariamente uma relação de complexidade crescente entre as questões e, por isto, a ordem recomendável de leitura do desenvolvimento dos componentes é: Porta Lógica XOR, Multiplexador, Extensor de Sinal, *Flip-Flop JK* e *Flip-Flop D*, Somador de 8 bits com Valor 4, Somador de 8 bits, Contador Síncrono, Banco de Registradores, Memória RAM, Memória ROM, Máquina de Estados Herbert, ULA 8 bits e, por fim Unidade de Controle 16 bits. O Software utilizado para realizar os testes foi o Software [Logisim](#), mais especificamente a versão 2.7.1.

2. COMPONENTES

2.1. REGISTRADOR FLIP-FLOP JK E FLIP-FLOP D

Figura 1 - *Flip-Flop JK*

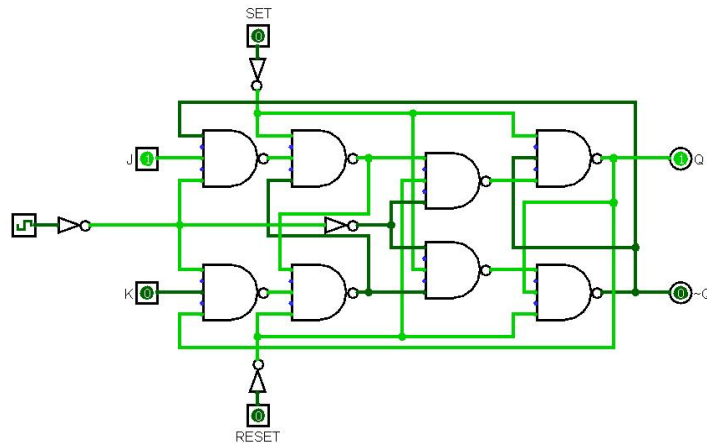
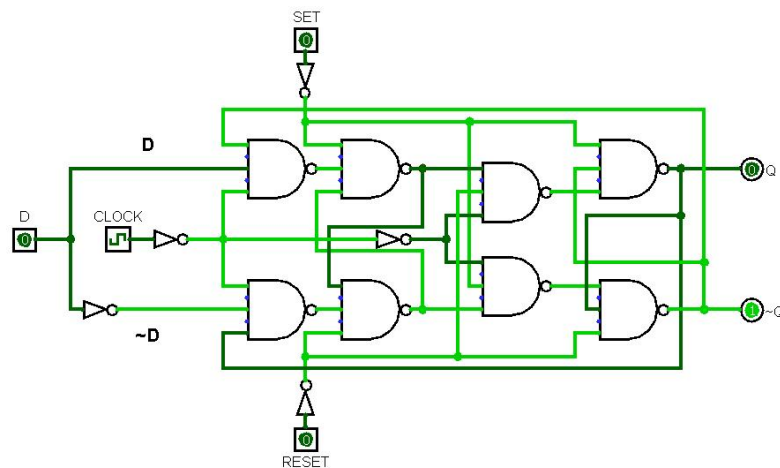


Figura 2 - *Flip-Flop D*



Os *Flip-Flops* são componentes da arquitetura de computadores que possuem em sua tabela-verdade uma característica destoante de outros componentes, a presença do último valor de saída efetuado pelo componente como próximo valor de saída em determinados casos, mesmo sem conhecimento do valor específico do mesmo, ou seja, desta forma é possível armazenar valores anteriores e trabalhar com os mesmos, podendo alterná-los, negá-los ou mantê-los para o funcionamento de demais circuitos.

2.1.1. Descrição pinos e lógica

O *Flip-Flop JK* possui quatro entradas: J, K, Preset e Clear, fora o Clock, sendo as entradas Preset e Clear correspondentes às respectivas entradas SET e RESET na Figura 1, além disto possui dois valores de saída: Q e ~Q. Por outro lado o *Flip-Flop D* possui apenas três entradas D, Preset e Clear, fora o Clock, e, de forma parecida com o componente *Flip-Flop JK* desenvolvido, as entradas Preset e Clear são correspondentes às respectivas entradas SET e RESET na Figura 2, e também possui dois valores de saída: Q e ~Q. O modelo escolhido de *Flip-Flop JK* e *Flip-Flop D* foi o modelo Chefe-Servente, que divide o

Flip-Flop em duas partes, como visto na Figura 3, onde existem dois *Flip-Flop* RS, uma versão mais simples de *Flip-Flop*, conectados, onde o Clock do *Flip-Flop* Servente é negado, impedindo o erro no caso de entradas 1 e 1, originalmente presente em *Flip-Flops* RS, como é demonstrado na Tabela 1.

Tabela 1 - Tabela-Verdade Flip-Flop RS Completa

R	S	Qant	Qs
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	1
0	1	0	0
0	1	1	0
1	1	0	ERRO
1	1	1	ERRO

2.1.2. Testes do componente

Ao realizar os testes foi possível gerar as tabelas-verdade do circuito do *Flip-Flop* JK e do circuito do *Flip-Flop* D e, após isto, simplificá-lo, como pode ser visto nas tabelas Tabela 2, Tabela 3, Tabela 4 e Tabela 5, onde o valor Qant está relacionado ao valor de Q anterior e Qs está relacionado saída do valor de Q atual.

Tabela 2 - Tabela-Verdade Flip-Flop JK Completa

J	K	Qant	Qs
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	1
0	1	0	0
0	1	1	0
1	1	0	1
1	1	1	0

Tabela 3 - Tabela-Verdade Flip-Flop JK Simplificada

J	K	Qs
0	0	Qant
1	0	1
0	1	0
1	1	~Qant

Tabela 4 - Tabela-Verdade Flip-Flop D Completa

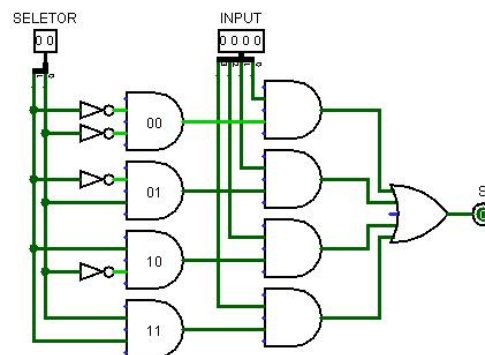
D	Qant	Qs
0	0	0
0	1	0
1	0	1
1	1	1

Tabela 5 - Tabela-Verdade Flip-Flop D Simplificada

D	Qs
0	0
1	1

2.2. MULTIPLEXADOR COM 4 OPÇÕES DE ENTRADA

Figura 3 - Multiplexador



O multiplexador é um componente extremamente comum em arquiteturas de computadores, pois é responsável por controlar o fluxo de dados em circuitos digitais.

2.2.1. Descrição pinos e lógica

O multiplexador possui duas entradas, INPUT e o SELETOR, possuindo, respectivamente 4 e 2 bits cada, além disto possui uma saída, S. O componente é responsável pelo desvio de fluxo em um circuito, ou seja, o valor de cada saída em um multiplexador será igual ao input apenas caso o valor do seletor desvie a corrente no sentido do mesmo.

2.2.2. Testes do componente

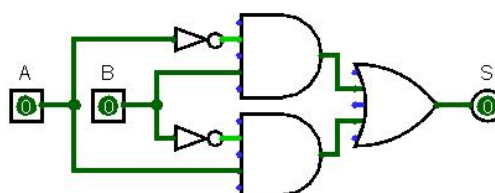
Ao realizar os teste com o multiplexador foi possível gerar um tabela-verdade como pode ser visto nas tabelas Tabela 6, sendo “X”, “Y”, “Z” e “W” os valores dos bits de INPUT, que variam de acordo com a entrada do dada pelo funcionamento do circuito.

Tabela 6 - Tabela-Verdade Multiplexador

INPUT	SELETOR	S
XYZW	00	X
XYZW	01	Y
XYZW	10	Z
XYZW	11	W

2.3. PORTA LÓGICA DO XOR

Figura 4 - Porta Lógica Xor



O componente *Xor* é um componente comumente utilizado em diversas arquiteturas de circuitos digitais, pois o mesmo obriga que apenas um dos valores seja verdadeiro para que a saída também seja.

2.3.1. Descrição pinos e lógica

O componente *Xor*, também conhecido como “ou exclusivo” funciona com a junção de dois componentes do tipo E, onde as entradas são negadas de forma alternada e seus resultados após o componente E são conectados à um único componente OU, desta forma, OU o primeiro valor (neste caso, o input A) é igual a 1 e o segundo valor (neste caso, o input B) é igual a 0, OU o primeiro valor é igual a 0 e o segundo valor é igual a 1, para que a saída final seja verdadeira.

2.3.2. Testes do componente

A tabela verdade do componente XOR já era previamente conhecida, pois o mesmo é utilizado em diversas outras arquiteturas, como é demonstrado por [Metrópole Digital](#), na publicação “[Circuitos Exclusive-OR \(XOR\) e Exclusive-NOR \(XNOR\)](#)”, e, após teste, foi possível obter os mesmos resultados, como é demonstrado na Tabela 8.

Tabela 7 - Tabela-Verdade XOR

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

2.4. SOMADOR 8 BITS COM VALOR 4

Figura 5 - Circuito Completo Somador 8 bits

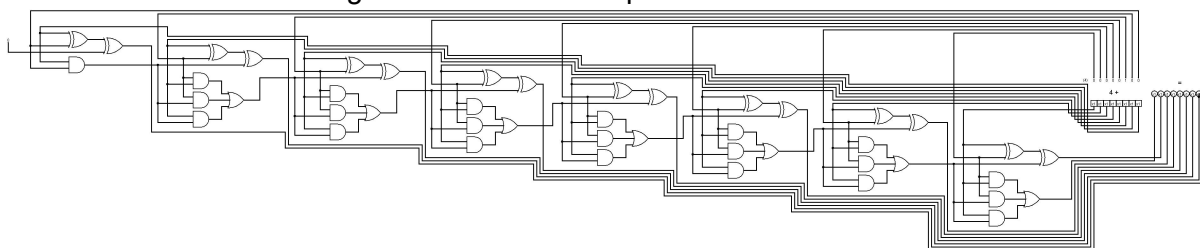
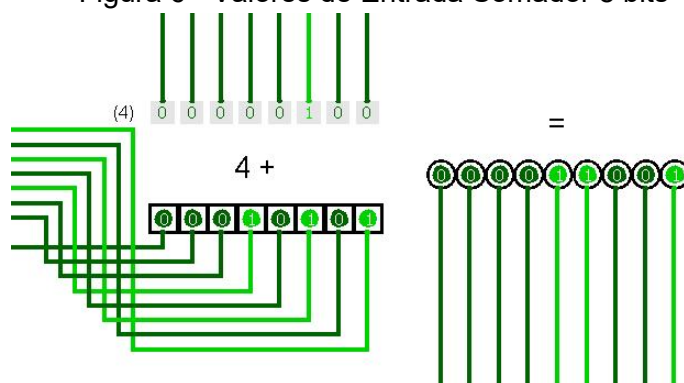


Figura 6 - Valores de Entrada Somador 8 bits



O Somador é um componente aritmético que soma valores binários utilizado para efetuar cálculos pelo computador, sendo que o seu tamanho varia de acordo com a quantidade de bits dos valores de entrada.

2.4.1. Descrição pinos e lógica

O funcionamento do Somador de 8 bits pode ser entendido a partir do funcionamento de uma seção do mesmo, como pode ser visto na Figura 7, que possui três valores de entrada de 8 bits, A, B, e Cin, sendo A e B os valores de entrada atuais e Cin o valor excedente da soma da seção anterior, além disto possui duas saídas de 1 bit, S e Cout, onde o valor de S corresponde ao valor da soma e Cout corresponde ao valor excedente da soma atual. Neste circuito o componente XOR 1 impede que soma de dois valores iguais a 1 provenientes de A e B resultem em 1, pois 1+1 em binário, diferente do sistema decimal comumente utilizado, resulta em 10, não em 2, pois o valor 2 não existe no sistema binário, da mesma forma o componente XOR 2 impede que, caso o valor do XOR 1 seja igual a 1 e o valor de Cin seja igual a 1, que o valor de S seja igual a 1. O Grupo de E determina o valor de Cout, de forma que caso o valor de A e B sejam iguais a 1, ou caso A ou B sejam iguais a 1 e Cin seja igual a 1, então o Cout será igual a 1, como pode observado na Tabela 9.

Figura 7 - Secção de Circuito Somador 8 bits

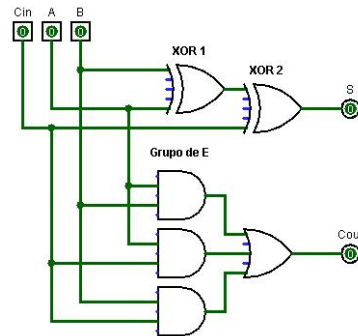


Tabela 8 - Tabela-Verdade Secção de Circuito Somador 8 bits

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2.4.2. Testes do componente

O circuito do Somador de 8bits é formado por diversas secções de somador, presente na Figura 7, encadeadas, como demonstrado na Figura 5, de forma que 8 valores de 1 bit de entrada (ou uma única entrada de 8 bits) e o valor fixo de 100 (que equivale ao valor 4 em binário), são somados, como demonstrado na Figura 6.

2.5. MEMÓRIA ROM

Figura 8 - Circuito Completo Memória ROM

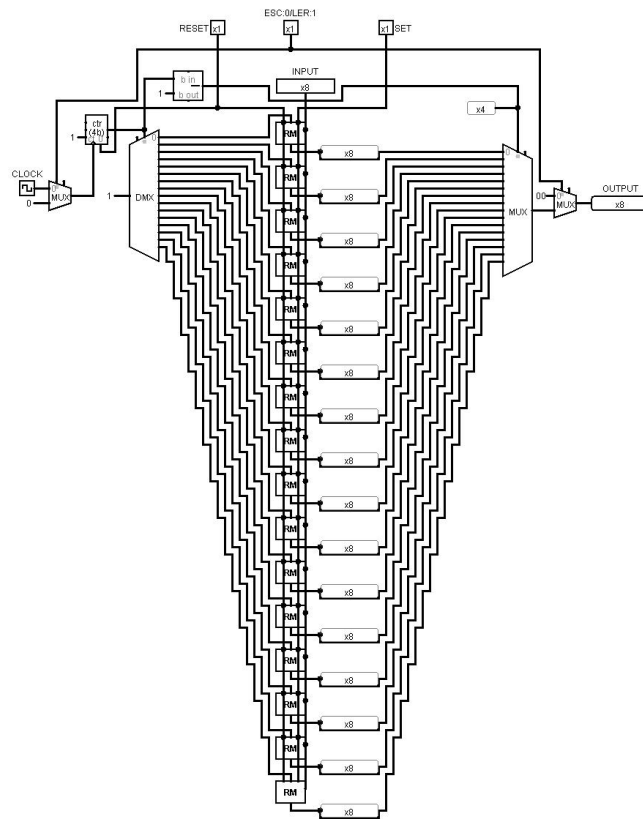


Figura 9 - Entradas e Saídas Memória ROM

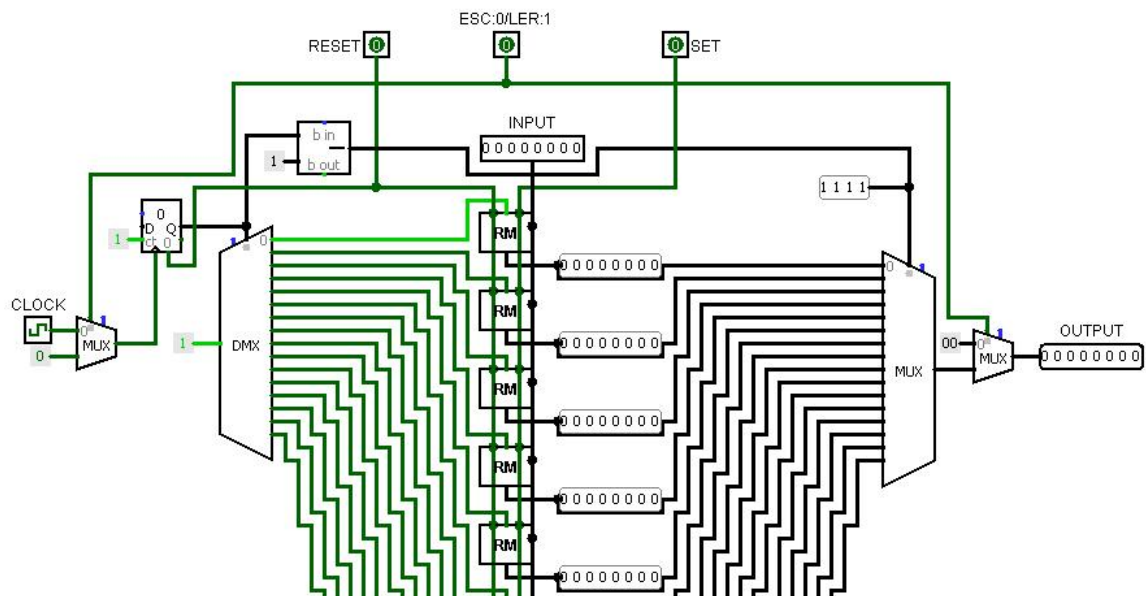
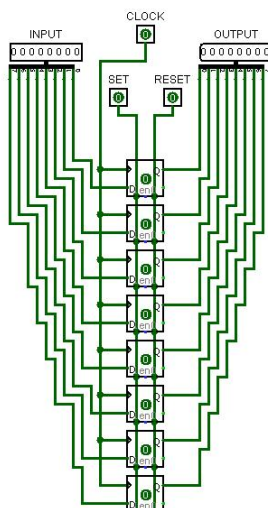


Figura 10 - Valor de Memória



A Memória ROM é uma memória que guarda instruções de forma sequencial, gerando algo parecido com um “histórico”, conceito comum em outras tecnologias, de forma que as instruções anteriores possam ser consultadas.

2.5.1. Descrição pinos e lógica

A Memória ROM, que possui o circuito presente na Figura 8, é uma memória sequencial, ou seja, guarda os valores de entrada em sequencia, neste caso uma pseudo-instrução de 8 bits foi adotada, com 16 espaços de memória disponíveis. O fluxo do circuito se inicia na entrada Ler/Esc, presente na Figura 9, que determina se a operação será de leitura ou escrita de valores na memória, em caso de escrita, o fluxo pode ser observado iniciando no multiplexador conectado com o Clock, que, caso a entrada Ler/Esc seja 0, o Contador Síncrono de 4 bits é incrementado, determinando o seletor no desmultiplexador com entrada constante 1, armazenando o valor de INPUT no Valor de Memória, representado na Figura 10, correto, salvando bit a bit em cada *Flip-Flop D* presente no mesmo. Caso a entrada Ler/Esc seja igual a 1, o valor atual do Contador Síncrono é decrementado em 1, enviando para o valor de saída OUTPUT a última instrução registrada.

2.5.2. Testes do componente

Após testes um problema de escrita de valores pôde ser percebido, pois o valor de entrada do principal do desmultiplexador, assim como o seu Seletor eram energizados ao mesmo tempo, gerando um erro no programa e salvando o mesmo valor em dois Valores de Memória diferentes, no entanto, ao energizar apenas o

seletor do desmultiplexador com o valor do Contador Síncrono e atribuindo a entrada principal o valor fixo 1.

2.6. MEMÓRIA RAM

Figura 11 - Memória RAM

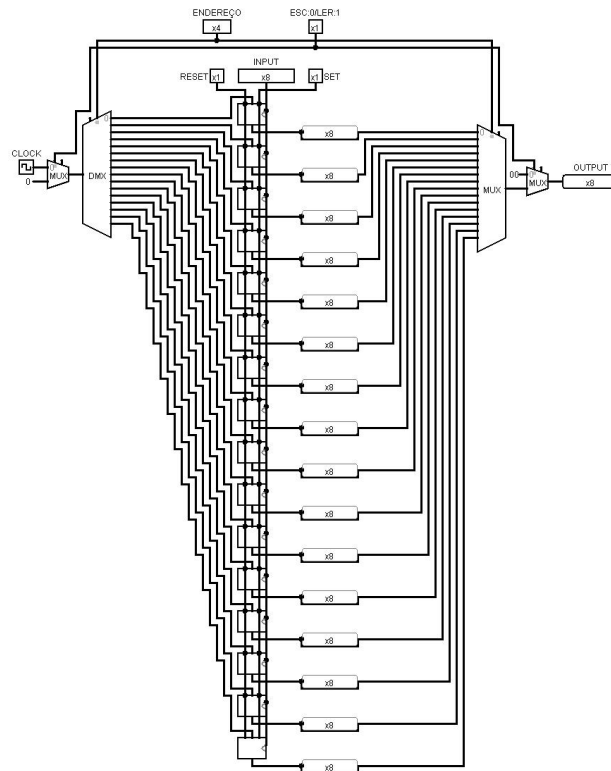
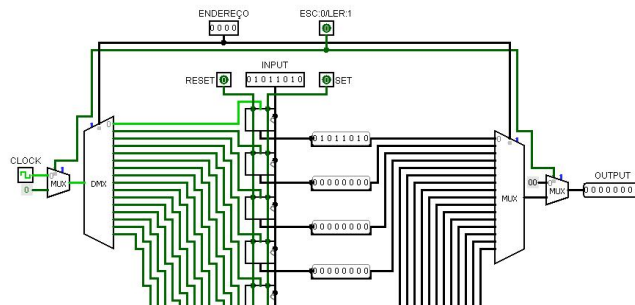


Figura 12 - Entradas e Saídas Memória RAM



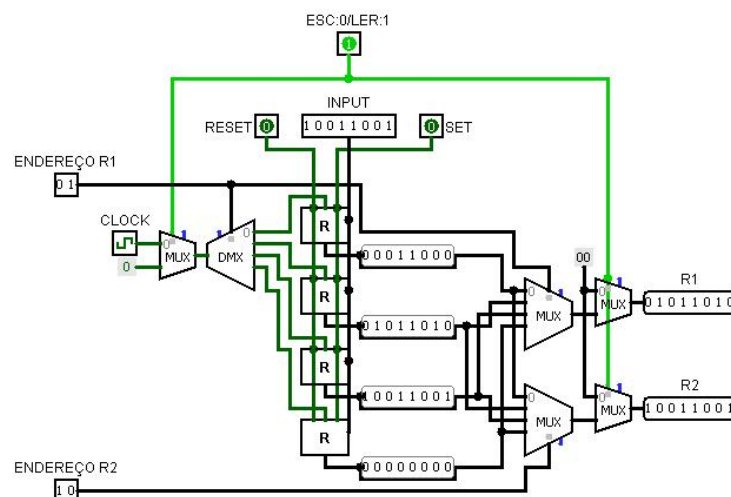
Diferente da Memória ROM, a Memória RAM é uma memória que guarda instruções de forma não-sequencial, ou seja, o local onde o valor será armazenado é designado pela entrada de Endereço.

2.6.1. Descrição pinos e lógica

A Memória RAM é uma memória não-sequencial, ou seja, guarda os valores de entrada de acordo com o endereço que lhe é atribuída, neste caso um valor de 8 bits foi adotado, com 16 espaços de memória disponíveis. O fluxo do circuito se inicia na entrada Ler/Esc, presente na Figura 12, que determina se a operação será de leitura ou escrita de valores na memória, em caso de escrita, o fluxo pode ser observado iniciando no multiplexador conectado com o Clock, que, caso a entrada Ler/Esc seja 0, o valor de Clock é atribuído ao demultiplexador e o sinal é enviado ao endereço determinado pela entrada Endereço, armazenando o valor de INPUT no Valor de Memória, presente na Figura 10 designado. Caso a entrada Ler/Esc seja igual a 1 o valor de Endereço é atribuído ao seletor do multiplexador conectado com os Valores de Memória, que irá se conectar com multiplexador que conectado com o OUTPUT, que irá selecionar a saída da memória por conta da entrada Ler/Esc, que é conectada ao seu seletor e irá enviar o valor para a saída OUTPUT.

2.7. BANCO DE REGISTRADORES

Figura 13 - Banco de Registradores



O Banco de Registradores armazena valores para serem utilizados em tempo de execução, realizando operações lógicas e aritméticas com os mesmos.

2.7.1. Descrição pinos e lógica

O Banco de Registradores, representado na Figura 13, tem seu funcionamento parecido com a Memória RAM, usando de demultiplexadores e multiplexadores, com o auxílio de entradas de endereço para armazenar valores em

Registradores, que possuem o mesmo circuito do Valor de Memória, presente na Figura 10, sendo sua maior diferença a quantidade de entradas e saídas, pois o mesmo possui duas entradas de endereço, ENDEREÇO R1 e ENDEREÇO R2, ambos com 2 bits, uma entrada de valor, INPUT e uma entrada de controle, Ler/Esc, além disto possui duas saídas de valores, R1 e R2. Quando o valor de Ler/Esc é igual a 0 o valor de ENDEREÇO R1 é utilizado para selecionar o Registrador para ser salvo o valor de INPUT, por outro lado caso o valor de Ler/Esc seja igual a 1, o valor da saída R1 será igual ao valor do Registrador no endereço da entrada ENDEREÇO R1, da mesma forma o valor da saída R2 será igual ao valor do Registrador no endereço da entrada ENDEREÇO R2.

2.8. SOMADOR 8 BITS

Figura 14 - Circuito Completo Somador 8 bits

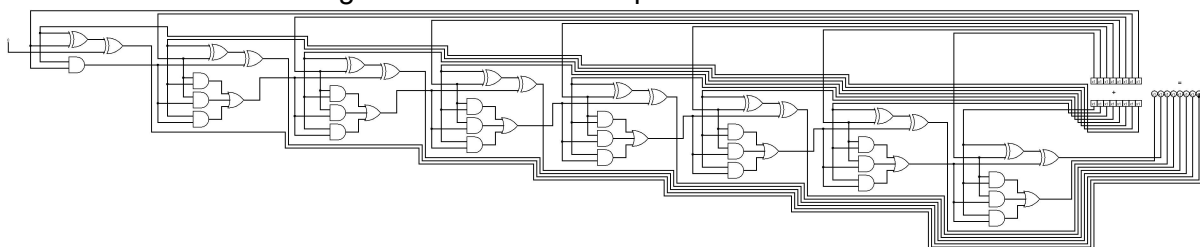
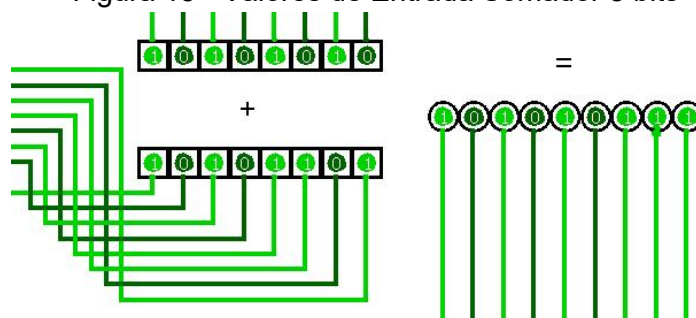


Figura 15 - Valores de Entrada Somador 8 bits



Assim como no ponto 2.4, que também se trata de um Somador 8 bits, o Somador 8 bits de dois valores quaisquer de 8 bits é um componente aritmético que soma valores binários utilizado para efetuar cálculos pelo computador.

2.8.1. Descrição pinos e lógica

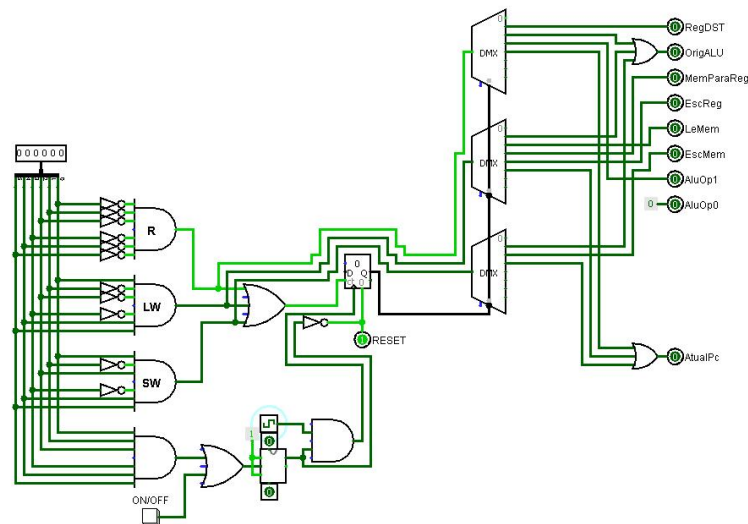
Da mesma forma que no ponto 2.4.1 o funcionamento do Somador de 8 bits pode ser entendido a partir do funcionamento de uma secção do mesmo, possuindo como único diferencial neste caso que não há valores fixos de entrada, podendo somar dois valores de 8 bits quaisquer.

2.8.2. Testes do componente

O circuito do Somador de 8bits é formado por diversas secções de somador, presente na Figura 7, encadeadas, como demonstrado na Figura 14, onde dois valores de 8 bits quaisquer são somados, como demonstrado na Figura 15.

2.9. UNIDADE DE CONTROLE 16 BITS

Figura 16 - Unidade de Controle 16 bits

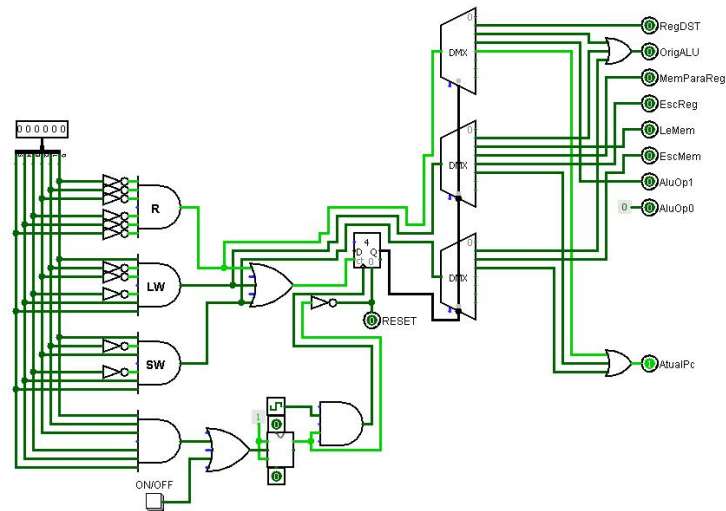


Unidade de Controle de 16 bits ordena os valores de entradas para o controle de outros componentes de um Processador de 16 bits.

2.9.1. Descrição pinos e lógica

A unidade de Controle 16 bits funciona a partir da saída de valores de “*flags*” periódicos de acordo com o valor de entrada de código de instrução. As *flags* são trilhas que são responsáveis por controlar outros componentes no sistema, desta forma, cada classe de instrução possui diferentes ordens de diferentes *flags* para o seu ideal funcionamento.

Figura 20 - Unidade de Controle 16 bits Classe R Caso 4



Após os testes da Unidade de Controle 16 bits, presentes nas figuras Figura 17, Figura 18, Figura 19 e Figura 20, realizando uma instrução de classe R foi possível perceber que as *flags* RegDST, OrigALU, AluOp1 e AtualPC foram utilizadas.

2.10. ULA 8 BITS

Figura 21 - ULA 8 bits

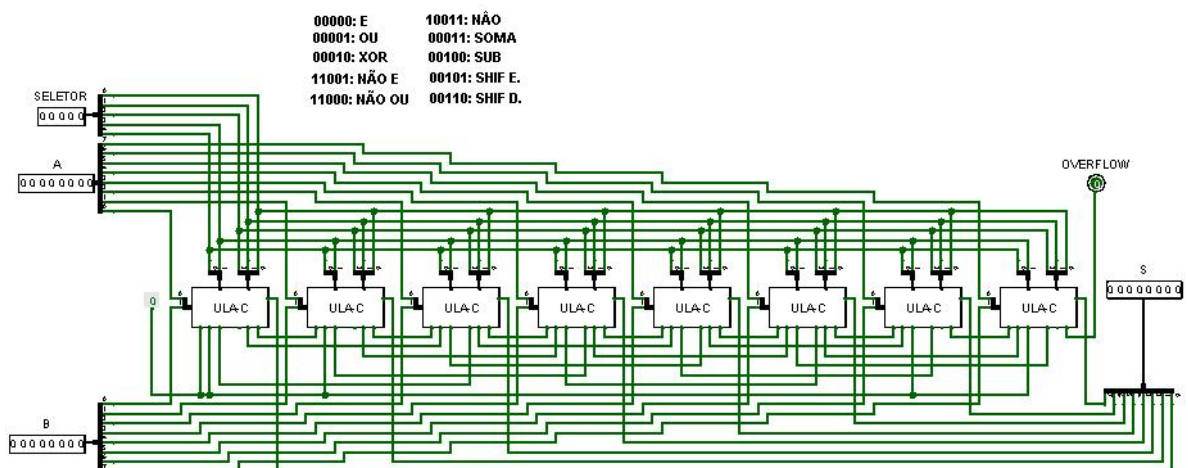
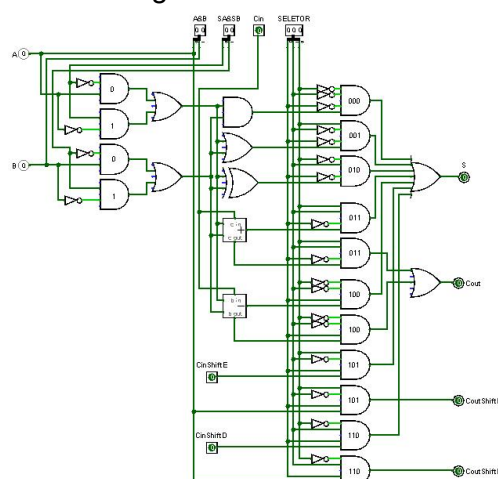


Figura 22 - ULA 1 bit



A ULA 8 bits, Figura 21, é formada por 8 ULAs de 1 bit, Figura 22, de forma que o valor final de OUTPUT é formado pelo conjunto de valores calculados por cada ULA de 1 bit formando um valor de saída de 8 bits.

2.10.1. Descrição pinos e lógica

A ULA de 1 bit possui oito valores de entrada, A, B, SA, SB, Cin, CinShiftE, CinShiftD, que possuem 1 bit e SELETOR, que possui 3 bits. As entradas A, B, SA, SB correspondem aos bits de entrada, sendo A e B os bits para cálculo, SA e SB para selecionar se os valores A e B serão negados, as entradas Cin, CinShiftE, CinShiftD são entradas de controle aritmético, sendo que seus valores advem de outras ULAs de 1 bit, e, por último o SELETOR é responsável por selecionar a operação a ser realizada, além disto, a ULA de 1 bit possui 3 valores de saída, S, Cout, CoutShiftE e CoutShiftD, de forma que o valor de saída S corresponde ao valor final da operação aritmética, o valor Cout corresponde ao valor excedente da operação e os valores CoutShiftE e CoutShiftD, são utilizados para realizar o *Shift* de dados binários para esquerda e direita respectivamente.

2.10.2. Testes do componente

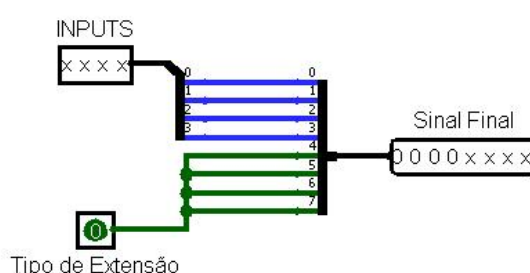
Tabela 9 - Códigos ULA

CÓDIGO	OPERAÇÃO
0 0 0 0 0	E
0 0 0 0 1	OU
0 0 0 1 0	XOR
1 1 0 0 1	NÃO E
1 1 0 0 0	NÃO OU
1 0 0 1 1	NÃO
0 0 0 1 1	SOMA
0 0 1 0 0	SUBTRAÇÃO
0 0 1 0 1	SHIFT ESQUERDA
0 0 1 1 0	SHIFT DIREITA

Ao estruturar as ULAs de 1 bit é possível passar uma única entrada de SELETOR que servirá para todas as ULAs de 1 bit e dividir bit a bit os valores de A e B e, após isto, agrupar os valores em uma única saída OUTPUT. Os códigos correspondentes as operações aritméticas estão relacionados na Tabela 10.

2.11. EXTENSOR DE SINAL 4 PARA 8 BITS

Figura 23 - Extensor de Sinal



O Extensor de Sinal de 4 para 8 bits é útil para transformar um valor previamente obtido para um tamanho maior de instrução, quando isto é necessário para o funcionamento do circuito, existindo dois tipos de extensão: extensão de valor 0 e extensão de valor 1, completando o bits excedentes com o valor do tipo de extensão.

2.11.1. Descrição pinos e lógica

O Extensor de Sinal, Figura 23, possui duas entradas Tipo de Extensão, que possui 1 bit e INPUTS, que possui 4 bits e apenas uma saída, Sinal Final, que possui 8 bits, de forma que os bits da entrada INPUTS ocupam os quatro últimos bits

do Sinal Final e o valor repetido da entrada Tipo de Extensão nos quatro primeiros bits do Sinal Final.

2.11.2. Testes do componente

O valor “XXXX” da entrada INPUTS significam que dependem do funcionamento do circuito em conexão com o extensor de sinal, desta forma os resultados do Extensor de Sinal serão “0000XXXX” se o valor de Tipo de Extensão for 0 e serão “1111XXXX” se o valor de Tipo de Extensão for 1.

2.12. MÁQUINA DE ESTADOS HERBERT

Figura 24 - Máquina de Estados Herbert

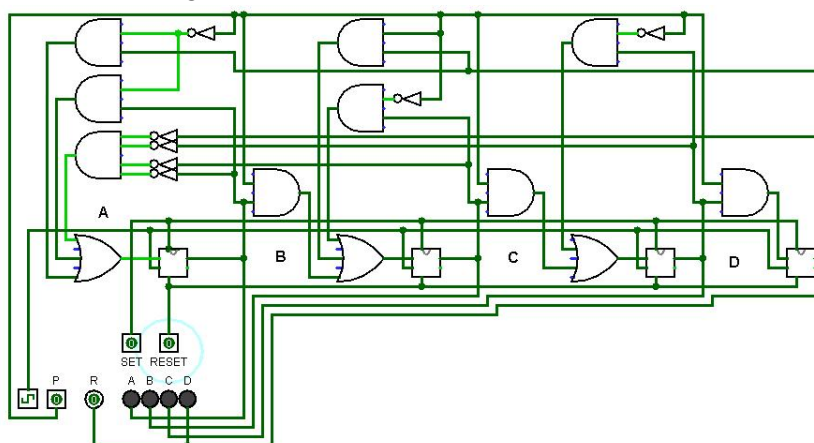
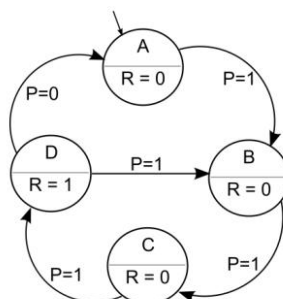


Figura 25 - Conceito Máquina de Estados Herbert



Uma Máquina de Estados é, resumidamente, uma máquina que devolve determinados valores dependendo do momento e se move de diferentes formas dependendo de uma determinada entrada. Esta Máquina de Estados não é nomeada, por isto lhe nomeei “Maquina de Estados Herbert”.

2.12.1. Descrição pinos e lógica

A Máquina de Estados, presente na Figura 24, é baseada no conceito de máquina de estados da Figura 25, desta forma pressupõe-se as verdades: “Caso

não haja estado atual, A torna-se o estado atual”, “Se o estado atual for A e $P = 1$, o estado atual torna-se B”, “Se o estado atual for B e $P = 1$, o estado atual torna-se C”, “Se o estado atual for C e $P = 1$, o estado atual torna-se D”, “Se o estado atual for D e $P = 1$, o estado atual torna-se B”, “Se o estado atual for D e $P = 0$, o estado atual torna-se A”, “Se A, ou B, ou C forem verdadeiro e $P =$, o determinado valor verdadeiro mantém-se verdadeiro”, “Se o estado atual for diferente de D, então $R=0$ ” e, por último, “Se o estado atual for D, então $R=1$ ”.

2.12.2. Testes do componente

Figura 26 - Máquina de Estados Herbert (Caso 1)

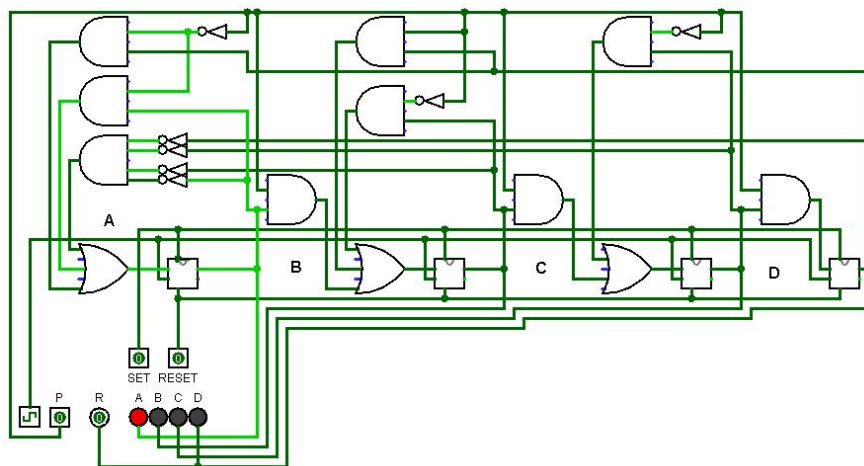


Figura 27 - Máquina de Estados Herbert (Caso 2)

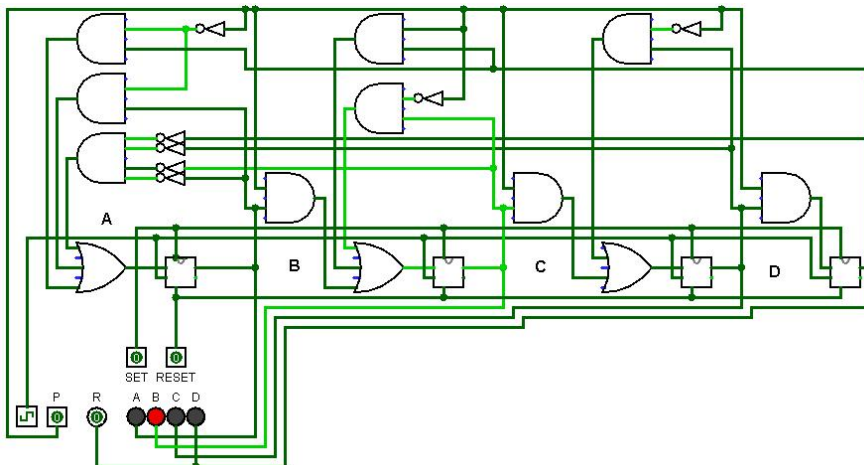


Figura 28 - Máquina de Estados Herbert (Caso 3)

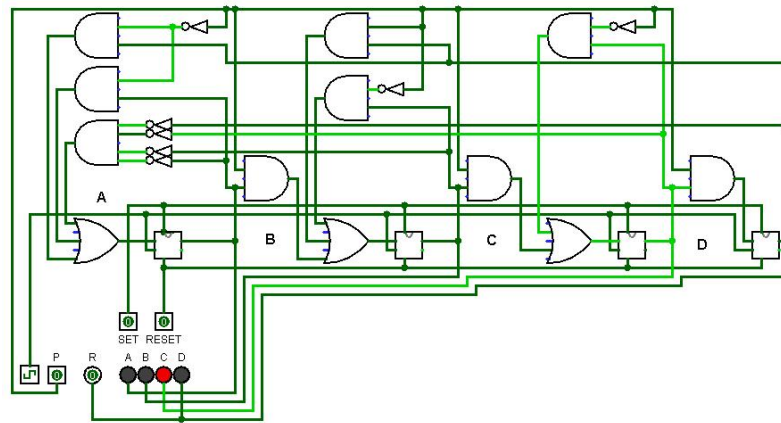


Figura 29 - Máquina de Estados Herbert (Caso 4)

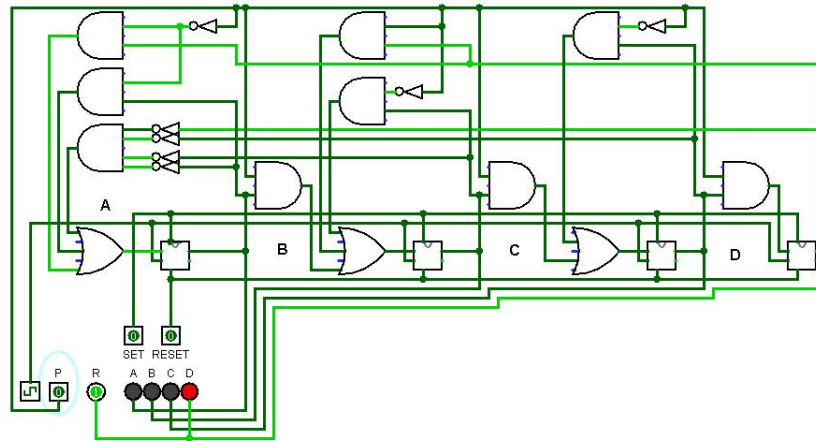
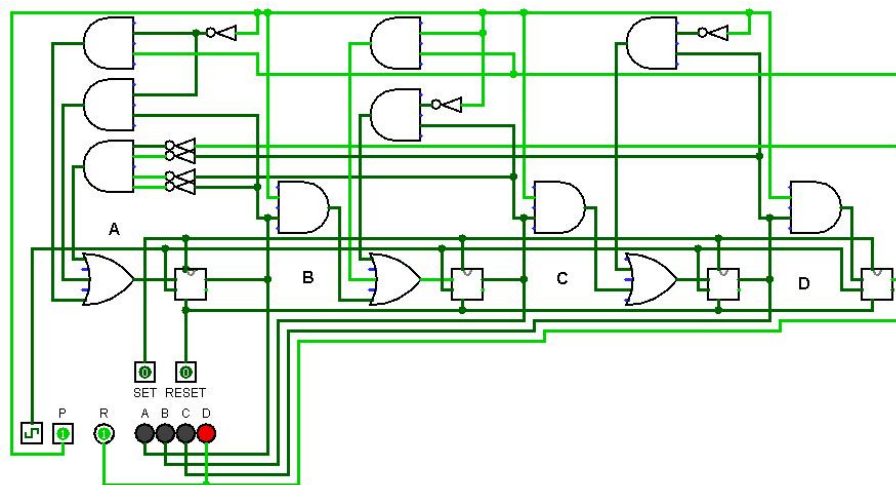


Figura 30 - Máquina de Estados Herbert (Caso 5)



O Contador Síncrono é um componente que realiza uma contagem bit a bit de um valor com a quantidade de bits delimitada pela quantidade de *Flip-Flops T* presentes em seu circuito.

2.13.1. Descrição pinos e lógica

Figura 32 - *Flip-Flop T*

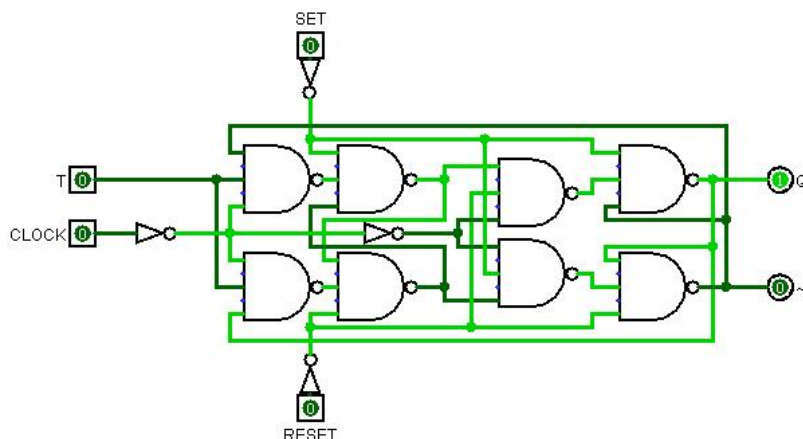


Tabela 11 - Tabela-Verdade *Flip-Flop T* Completa

T	Qant	Qs
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 12 - Tabela-Verdade *Flip-Flop T* Simplificada

T	Qs
0	Qant
1	\sim Qant

O contador Síncrono desenvolvido possui 4 bits e, consequentemente, quatro *Flip-Flops T*, presente na Figura 32, demonstrado na Figura 31, que, após ser desenvolvido e testado gerou a tabela-verdade Tabela 11 e, após simplificação, gerou a Tabela 12, desta forma o contador atribui o valor 1 a entrada do *Flip-Flop T* apenas quando todos os valores anteriores são iguais a 1, alternando-o, com exceção do último *Flip-Flop T*, que sempre alterna o seu valor.

2.13.2. Testes do componente

Os valores do Contador Síncrono são obtidos pela propriedade de alternância de valores anteriores do *Flip-Flop T*, de forma que é possível atribuir entradas iguais a 1 quando valores também sejam iguais a 1, de forma parecida com um Somador.

3. CONSIDERAÇÕES FINAIS

Finalmente, agradeço ao professor Herbert pela oportunidade de estudar, entender e, pelo incrível que possa parecer, despertar um gosto em mim por hardware, assunto este que nunca gostei. Ao finalizar este relatório juro que, caso tire uma nota maior ou igual a 9 nesta disciplina, uma das minhas próximas tatuagens será em formato de um circuito.

4. REFERÊNCIAS

Canal Tell Moitas, <https://www.youtube.com/@tellmoitas9661>

Canal Newton Lago, <https://www.youtube.com/@newtonlago7665>

Site Filipe Flop, <https://www.filipeflop.com/blog/entendendo-o-flip-flops/>

Site Metr pole Digital, <https://materialpublic.imd.ufrn.br/>

Biblioteca UFRR, <https://ufrn.br/bibliotecas/>