UNIVERSIDADE FEDERAL DE RORAIMA CENTRO DE CIÊNCIA E TECNOLOGIA BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DCC301 – ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES (2022.2)

Data de entrega: 29/11/2022

DISCENTES:
FELIPE RUBENS DE SOUSA BORGES (2020020120)
WILLIAM MATHEUS DURANS SILVA (2020023928)

Arquitetura e Organização de Computadores

Relatório

Relatório técnico de acordo com a proposta feita em sala de aula pelo docente Prof. Dr. Herbert Oliveira relacionada a Implementação de Componentes com suas devidas características.

Boa Vista-RR
2022.2

Relatório técnico de Implementação de Componentes

Relatório apresentado como requisito parcial para obtenção de nota na disciplina de Arquitetura e Organização de Computadores, ofertada pelo curso de Ciência da Computação da Universidade Federal de Roraima.

Prof. Dr. Herbert Oliveira

Boa Vista-RR
2022.2

Resumo

Esse relatório consiste na apresentação sobre a implementação dos devidos 13 componentes solicitados em documento via Sigaa no dia 17 de outubro de 2022, expondo suas devidas descrições, funcionalidades, imagens de componentes, testes e descrição dos testes realizados via software.

O software utilizado para a elaboração da atividade foi o Logisim na sua versão 2.7.1.

Sumário

Lista de Figuras

Componente 1: Registrador Flip-Flop do tipo D e do tipo JK.

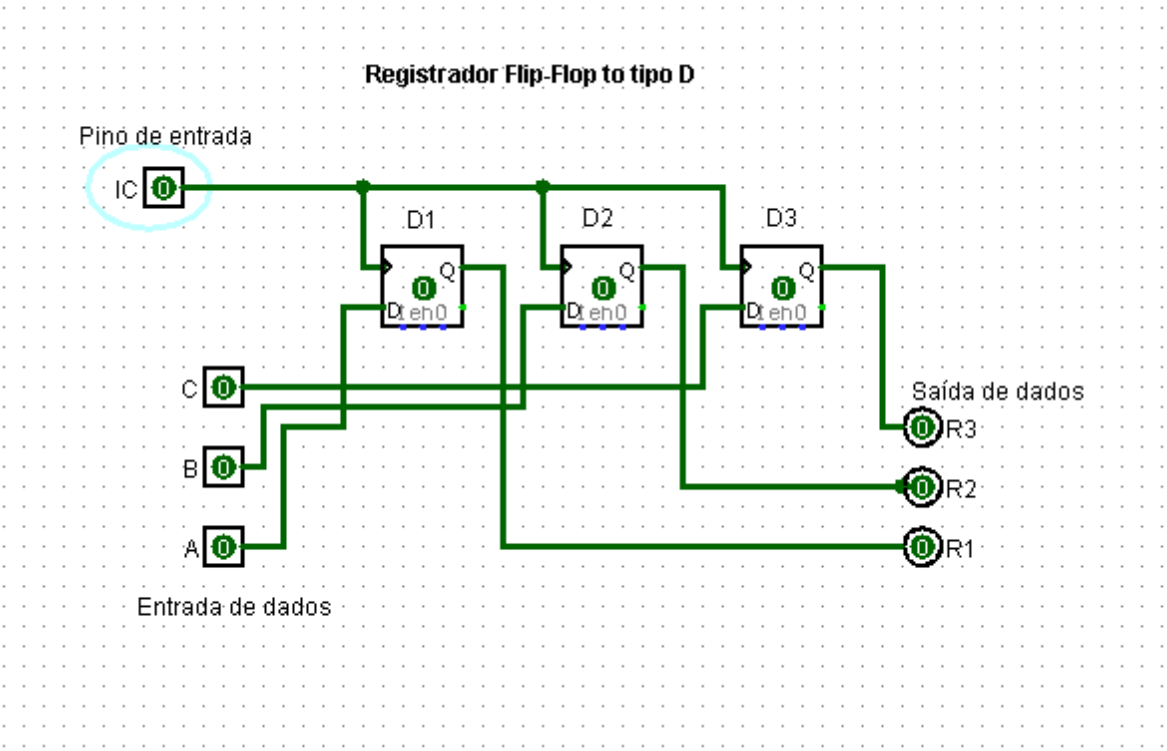
Para registrador Flip-Flop do tipo D:

Descrição dos pinos, lógica e funcionalidade

Para a estrutura do registrador Flip-Flop para o tipo D, utilizamos 6 pinos/componentes na elaboração do circuito, sendo esses divididos em:

- 1x Pino de entrada: pino responsável pelo impulso no clock do circuito, alternando o valor lógico entre 0 e 1 para os dados que estão guardados nos registrados.
- 1x Pino de entrada de dados: Este pino é responsável por armazenar os dados que são inseridos e serão lidos pelo circuito.
- 1x Pino de saída de dados: Este pino é responsável por apresentador o resultado do circuito logico, ou seja, assim que um impulso for dado no clock e os dados forem lidos e armazenados nos registradores, serão atualizados na saída de dados igualmente aos dados de entrada.
- 3x Flip-Flop do tipo D: Aqui estão os registradores do circuito integrado, mais especificamente, todos os três registradores do *tipo D*, que armazenam, atualizam e enviam os valores para a saída de dados do circuito durante cada impulso de clock que é executado no sistema.

Imagem dos componentes do circuito integrado



Resultado de testes

IC	C	B	A	R3	R2	R1
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1

Descrição dos testes

Durante a fase de *testes* os resultados foram obtidos com *sucesso*, pois todos os *dados* de entrada e saída foram atualizados com *êxito* a partir da atualização do impulso de clock que o pino de entrada aplicou ao circuito durante sua execução.

Para registrador Flip-Flop do tipo JK:

Descrição dos pinos, lógica e funcionalidade

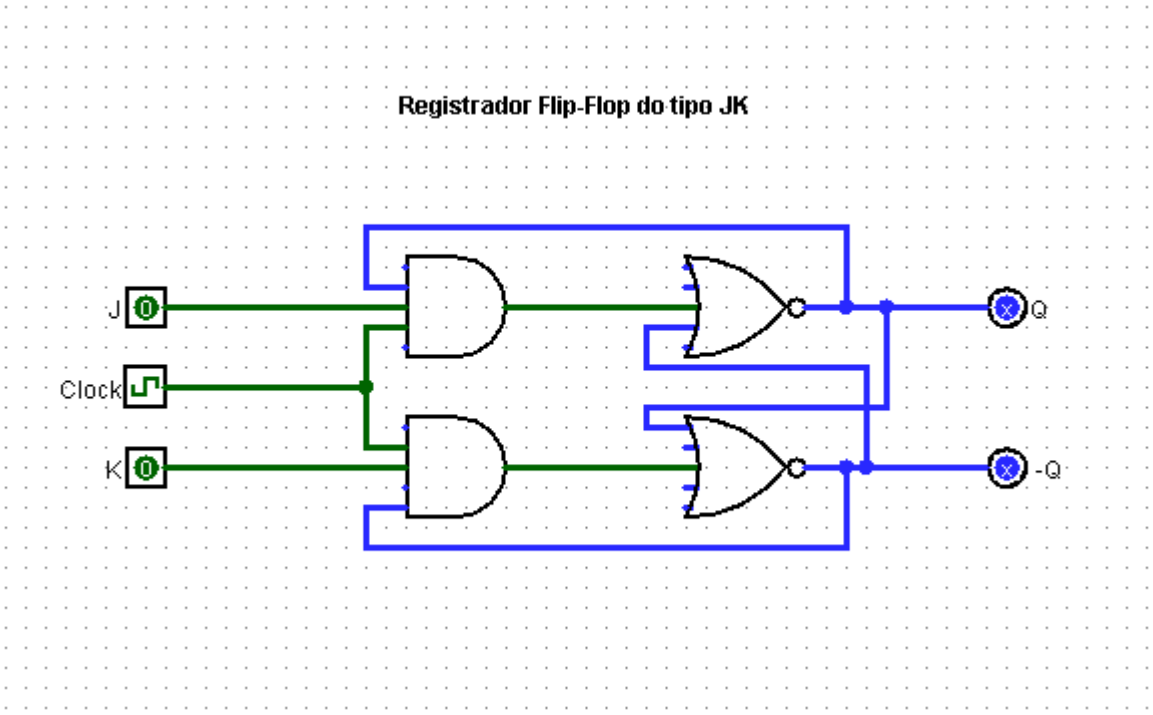
Para o registrador do FF-JK foram utilizados 9 pinos/componentes no total do circuito integrado, onde, a partir desses 9, temos 5 pinos distintos, cada um com sua atribuição e atividade, sendo eles:

- 2x pinos de entrada de dados: Esses são responsáveis por guardar e mostrar ao circuito quais dados estão sendo inseridos e estabelecidos ali.
- 2x Porta AND: Este pino é o responsável por verificar o valor lógico do operando, resultando em um valor *true* a ele atribuído somente se todos os operandos tiverem valor lógico verdadeiro.
- 2x Porta NOR: Essas portas são operadores booleanos que resultam na negação do valor operador, portanto, ele apenas tem resultado lógico igual a verdadeiro se ambos os operandos forem falsos.

- 2x pinos de saída de dados: Estes exibem o valor lógico de toda a operação realizada depois de um impulso de clock no circuito a partir da leitura do dado que foi armazenado na entrada de dados, passando por ambas as portas.

- 1x Clock: Este pino de clock é o responsável por dar o impulso do sistema para que as portas possam agir sobre o dado armazenado, expondo ao fim o resultado lógico daquele valor.

Imagem dos componentes do circuito integrado



Resultado de testes

J	K	Q	Q2
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

Descrição dos testes

Durante a fase de *testes* os resultados foram obtidos com *sucesso*, pois todos *saída* foram atualizados com *êxito* a partir da atualização do impulso de clock realizado pelo pino no circuito durante sua execução, pós leitura de ambas as portas de operantes.

Componente 2: Multiplexador de quatro opções de entrada.

Descrição pinos, lógica e funcionalidade

No circuito integrado de multiplexador com 4 opções de entrada, foram utilizados 9 pinos/componentes para a arquitetura da atividade, sendo eles dispostos da seguinte maneira:

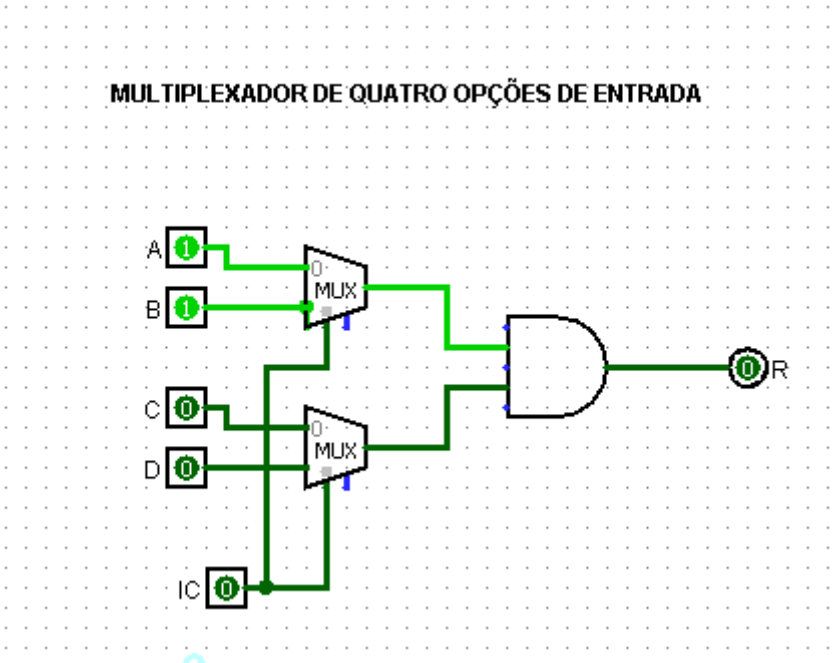
- 5x pinos de entrada de dados (A, B, C, D e IC): Dessa maneira, os pinos dispostos de A até D são os pinos que receberão os dados para a execução lógica do *multiplexador*, já o pino IC (Impulso no clock) será responsável pela atividade lógica dentro do *multiplexador*, passando então para a *porta AND* o resultado lógico correto a partir dos dados inseridos nos outros 4 pinos de entrada do circuito.

- 2x pinos de multiplexador: Esses pinos são os multiplexadores, um circuito dedicado a combinações que a partir do impulso de clock, lê os valores recebidos pelos demais pinos de entrada e aplica a saída desses dados para a *porta AND* do circuito, onde o resultado lógico a partir da análise combinatória será expresso.

- 1x pino de Porta AND: A porta AND é um componente de operação lógico que receba os resultados de dados dos multiplexadores e irá verificar se todos os operandos ali são um valor lógico verdadeiros.

- 1x pino de saída de dados: Este pino será responsável por apresentar o resultado lógico do circuito depois que todas as outras etapas supracitadas forem executadas dentro desse componente, exibindo assim o resultado lógico de acordo com a interpretação da porta AND a partir dos valores apresentados em ambos os multiplexadores.

Imagem dos componentes do circuito



Resultado de testes

A	B	C	D	IC	R
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	1

Descrição dos testes

Durante a fase de *testes* os resultados foram obtidos com *sucesso*, pois todos os *dados* de entrada foram atualizados após a execução do impulso de clock nos multiplexadores, passando pela interpretação de valor lógico da porta AND do circuito.

Componente 3: Porta lógica XOR usando port map com os componentes: AND, NOT, e OR.

Descrição pinos, lógica e funcionalidade

Para elaborar o circuito lógico utilizando o componente XOR, fizemos dois circuitos distintos a partir do port map, isso para mostrar a diferença que o componente do tipo NOT traz para a tabela verdade ao fim de um ciclo de clock no circuito. Foram utilizados no primeiro modelo 6 pinos/componentes, enquanto que no segundo modelo usamos 8 pinos/componentes, são eles:

- 2x pinos de entrada de dados: Esses pinos armazenam o dado na qual o circuito recebeu.
- 1x porta AND: Este é um componente de operação lógico que verifica se os operandos são valores lógicos do tipo *true* (verdadeiros).
- 1x porta OR: Este é um componente de operação lógico que resulta em um valor lógico do tipo *false* (falso) se, somente se, todos os operandos forem falsos.
- 1x porta XOR: Por fim, a última porta do circuito é uma porta de duas entradas que produz na saída o nível lógico 1 quando as entradas forem de valor lógico diferentes, e o nível lógico 0 quando as entradas forem de valores lógicos iguais.
- 1x pino de saída de dado: Este pino será responsável por mostrar o resultado lógico do circuito depois que todas as etapas acima forem executadas após um ciclo de clock ser realizado, apresentando o valor lógico entre os operandos.
- 2x portas NOT: Então, a última porta lógica do circuito é a porta do tipo NOT, ela é um operador do tipo negação, que informa o valor lógico contrario do valor real sobre o operando ao qual estiver conectada. Ela altera o resultados da tabela verdade como demonstrado e aplicado nas imagens 2 e 2.1 em relação as imagens 1 e 1.1 do circuito elaborado.

Imagem dos componentes do circuito

Imagem 1 (sem porta lógica NOT)

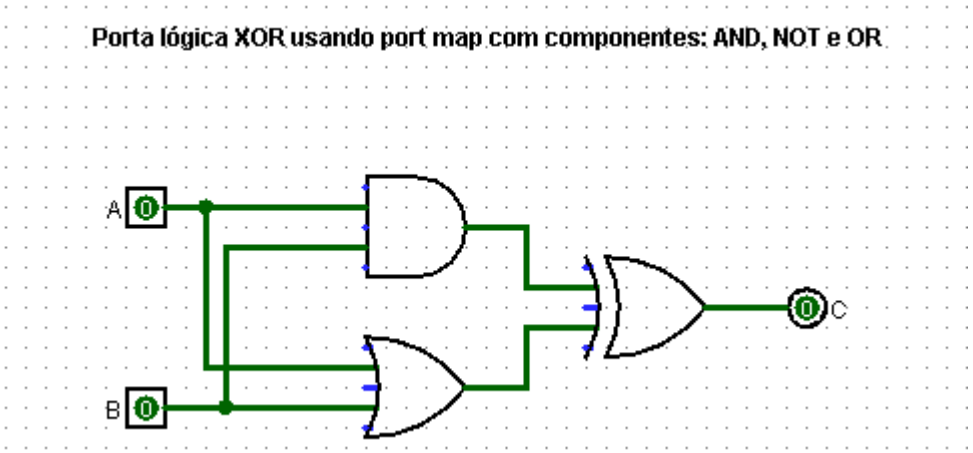
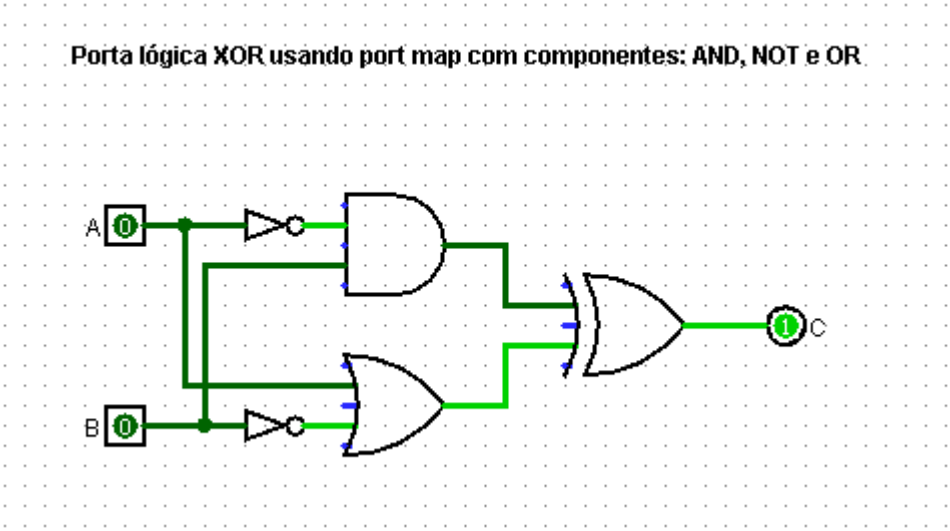


Imagem 2 (com porta lógica NOT)



Resultado de testes

Imagem 1.1 (tabela para o circuito da imagem 1)

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Imagem 2.1 (tabela para o circuito da imagem 2)

A	B	C
0	0	1
0	1	1
1	0	1
1	1	1

Descrição dos testes

Durante a fase de *testes*, toda a tabela verdade referente a este circuito foi atualizada com sucesso, tanto com a utilização de porta lógica do tipo NOT quanto sem as mesmas, como mostra as diferentes tabelas verdades de acordo com os circuitos apresentados.

Componente 4: Somador de 8 bits que recebe um valor inteiro e soma com o valor 4.

Descrição pinos, lógica e funcionalidade

Imagem dos componentes do circuito

Resultado de testes

Descrição dos testes

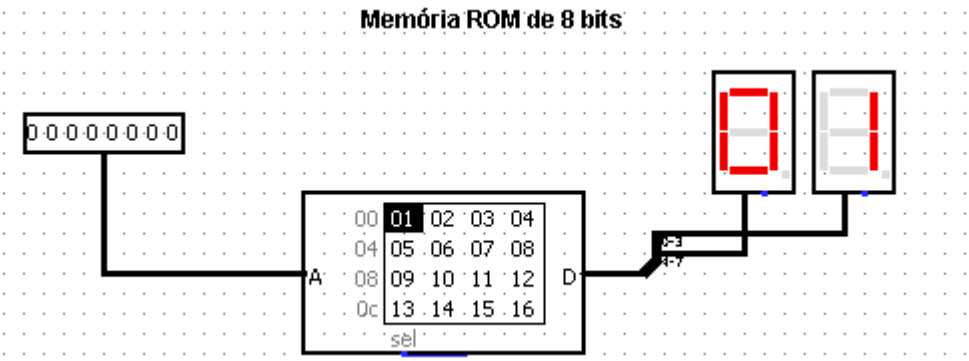
Componente 5: Memória ROM de 8 bits.

Descrição pinos, lógica e funcionalidade

No componente de Memoria ROM com 8 bits, utilizamos 5 componentes diferentes para demonstrar a funcionalidade do circuito, que acontece da seguinte maneira: Os bits de entrada vão ser setados pelo usuário dentro do campo de 8 bits distribuídos e o endereço digitado será localizado dentro da memória, exibindo nas placas de led o valor ali armazenado. Para este circuito, os componentes/pinos utilizados foram:

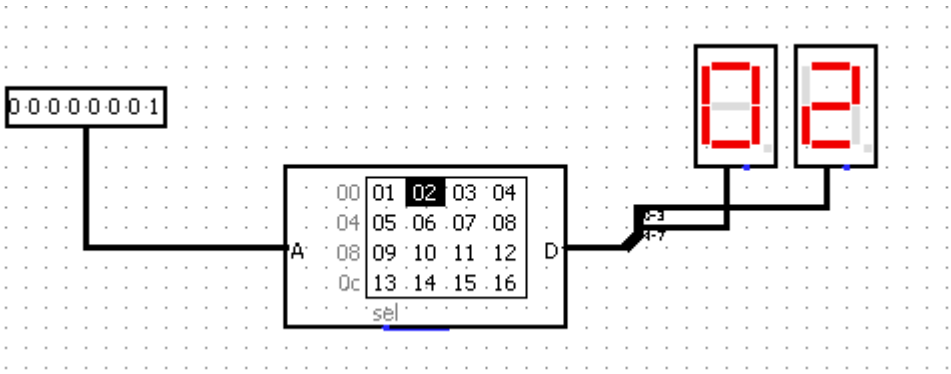
- 1x pino de entrada de dados: O pino utilizado tem suporte para 8 bits e receberá os dados do usuário para poder buscar o endereço solicitado.
- 1x Memória ROM de 8 bits: A memoria ROM vai armazenar diferentes valores para cada endereço de que ela pode armazenar (256 totais), a pra cada endereço chamado pela entrada de dados, o valor guardado passará para o distribuidor.
- 1x Distribuidor: O distribuidor de dados de 2 bits vai enviar os dados da memória pelos seus caminhos por onde está ligado.
- 2x Displays Hexadecimais: Os displays irão funcionar como um telão de led, onde ao receber os dados do distribuidor, mostrarão o valor que está ali guardado naquele endereço de memória da qual o usuário fez a chamada.

Imagem dos componentes do circuito

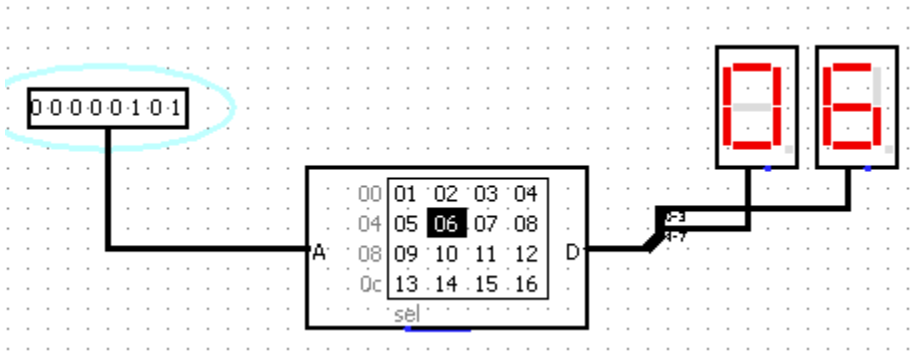


Resultado de testes

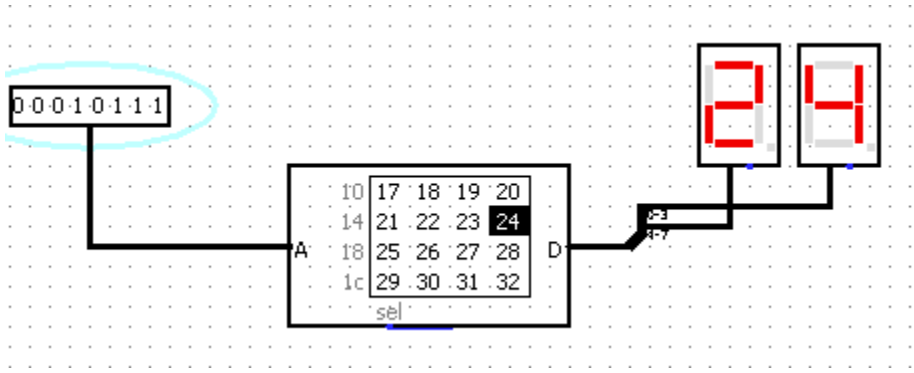
Teste 1



Teste 2



Teste 3



Descrição dos testes

Os testes funcionaram com *sucesso*, com apresentado nos exemplos acima, onde para cada endereço de 8 bits da memória, são apresentados nos displays hexadecimais os valores corretos com que estão guardados.

Obs: Apenas os 32 primeiros endereços da memória tiverem seus valores registrados e guardados para fins de testes, contudo, por ser um circuito de memória ROM de 8 bits, possui ao todo 256 endereços diferentes para serem trabalhados.

Componente 6: Memória RAM de 8 bits.

Descrição pinos lógica e funcionalidade

Na elaboração do circuito para memória *RAM* de 8 bits, dividimos o circuito em 4 partes para seu funcionamento dentro da seguinte proposta, poder buscar e localizar endereços dentro da memória, podendo também alterar o valor pertencente a esses endereços se, somente se, os *Buffers* presentes no circuito forem abertos para a memória *RAM*, liberando a atualização dos dados. Para esse circuito foram usados 29 pinos/componentes, que estão dispostos da seguinte maneira.

- 22x pinos de entrada de dados: Para estes casos, os pinos possuem duas finalidades para o circuito, a primeira será a de receber os endereços desejados para realizar a busca dentro da memória, a segundo será para permitir a passagem desses endereços percorrem o circuito (conexões com os *buffers*).

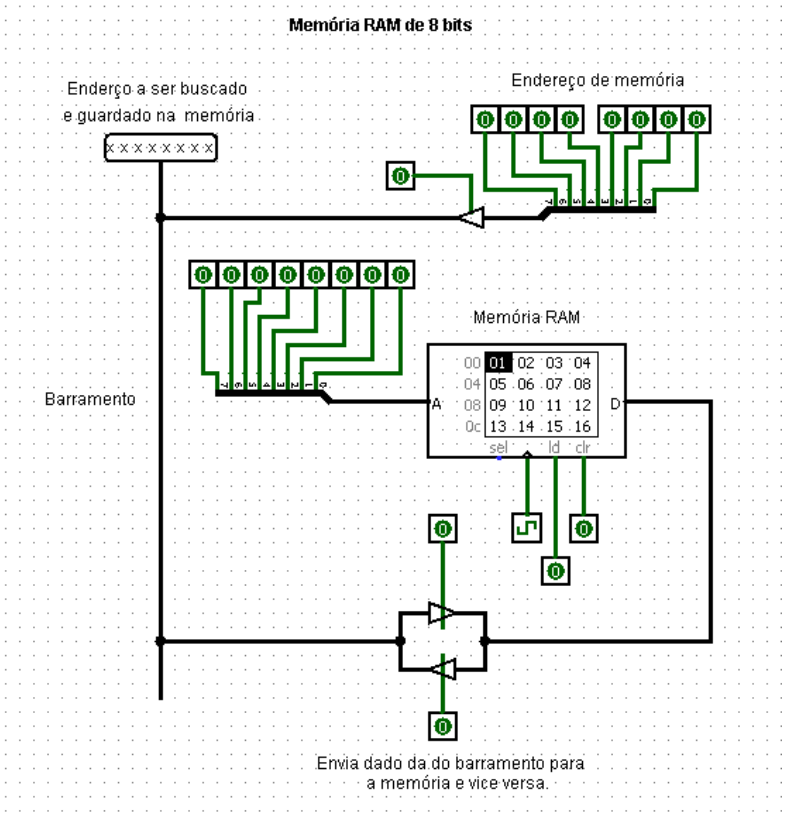
- 1x pino de saída de dados: Este pino setado com 8 bits será responsável por ler e interpretar o endereço solicitado, encontrando qual o valor nele está guardado e que alterações serão feitos (caso solicitadas), passando assim pelo *barramento* do circuito carregando essas informações diretamente para a memória.

- 2x distribuidores: Neste caso, os distribuidores são atarefados com o propósito de buscar e localizar endereços na memória, sejam para verificar o valor ali armazenado ou alterar o valor pertencente, a depender claro do que está sendo realizado pelo circuito.

- 3x buffers: Os buffers para o circuito são encarregados de encaminhar os dados informados nas entradas de endereço, apenas se os pinos a eles conectados estiverem em valor lógico 1 durante o ciclo de clock, autorizando assim o fluxo de dados no circuito, caso o valor lógico dos pinos a eles ligados estejam em 0, mesmo durante um impulso de clock, os dados não passarão pelo *barramento* do circuito, assim sendo, não serão passados os dados solicitados para buscar endereços ou alterar os valores conforme a chamada.

- 1x memória RAM: Por último, o principal componente do circuito, a memória RAM de 8 bits capaz de armazenar/guardar até 256 valores/dados diferentes no seu interior, permitindo que para cada um dos seus 256 espaços de locação, exista um endereço de memória único e distinto dos demais, de modo que cada um destes endereços pode ser chamado pela sua entrada A (entrada de dados da memória pra localizar e buscar seus endereços) ou chamado pela sua entrada D (que lê e escreve valores e/ou dados a serem guardados para o endereço que receber a chamada do usuário antes de passar pelo barramento).

Imagem dos componentes do circuito



Resultado de testes

Imagem 1

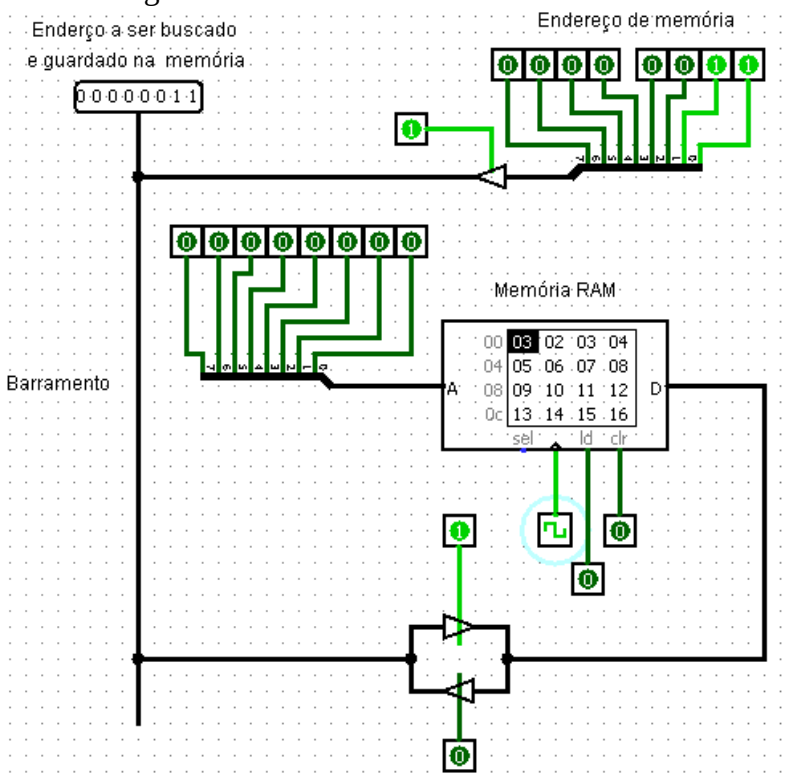


Imagem 2

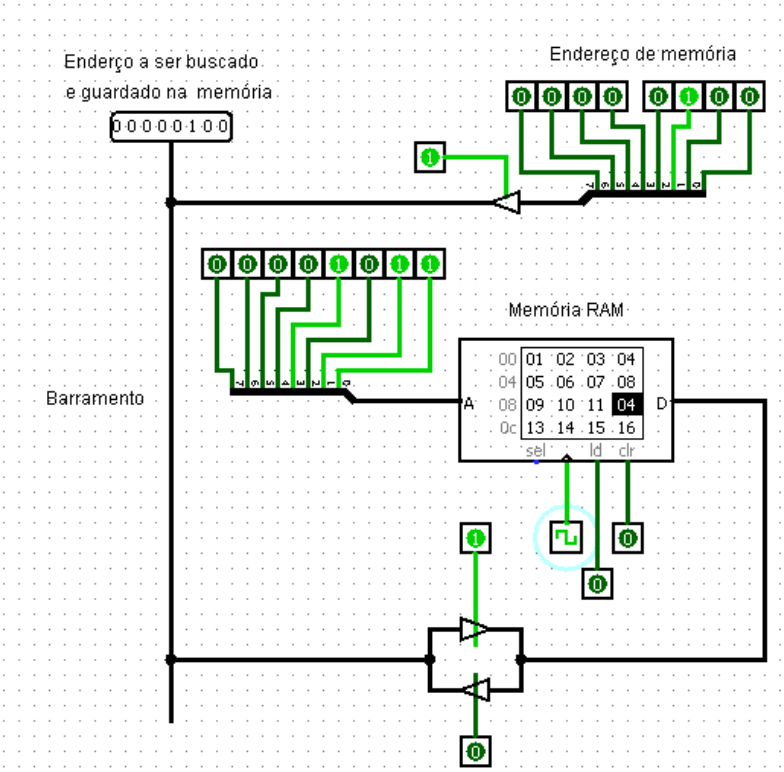
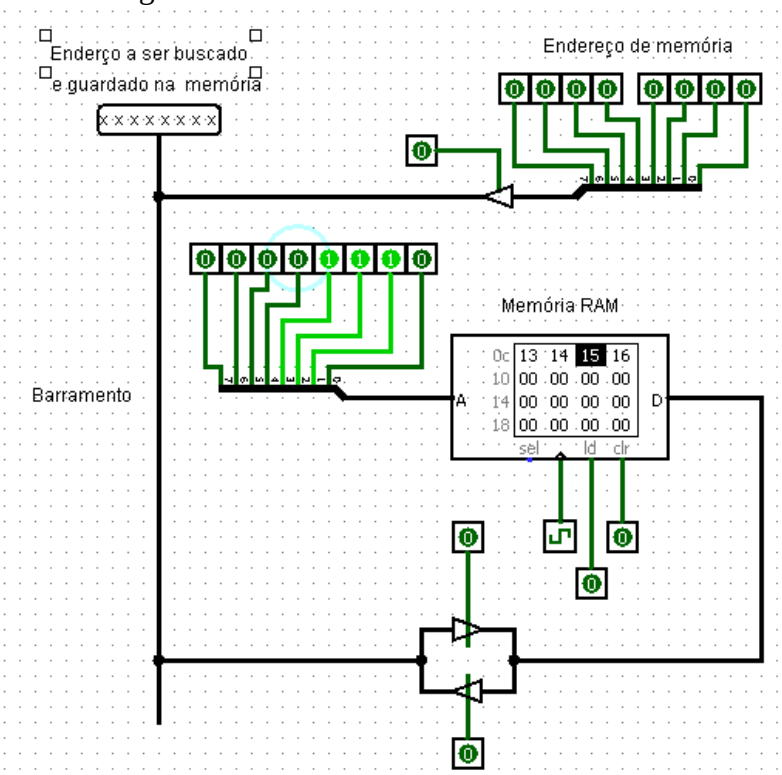


Imagem 3



Descrição dos testes

Os testes foram realizados com sucesso, conforme as imagens acima, de maneira que, para a imagem 1 temos o seguinte cenário: Foi setado um endereço de memoria (0000 0011), onde o mesmo foi impulsionado para o barramento do circuito, lido e depois impulsionado para dentro da memória, onde o endereço foi localizado, o valor foi reconhecido (03) e foi guardado no primeiro espaço da memória, pois o distribuidor no qual está ligado mostra o primeiro endereço da memória (0000 0000).

Conforme a imagem 2, temos um cenário similar ao da primeira imagem, de maneira que: O endereço do distribuidor passa de (0000 0000) para (0000 1011), que representa o espaço 12 dentro da memória, enquanto que o endereço de memória chamado e passado ao barramento foi (0000 0100), na qual guarda o valor 04 dentro da memoria RAM, quando este valor foi lido e jogado para a memória após um impulso de clock, perceba que o valor 04 (pertencente ao endereço chamada 0000 0100) foi guardado na posição 12 (posição dentro da memória registrada com o endereço 0000 1011).

Por fim, para a imagem 3 está apenas uma demonstração de como buscar por uma posição especifica dentro da memória para descobrir seu valor ali guardado, sem mesmo ter uma outra chamada na qual será passada pelo barramento para receber um impulso de clock.

Obs: Apenas os 16 primeiros endereços da memoria tiverem seus valores registrados e guardados para fins de testes, contudo, por ser um circuito de memoria RAM de 8 bits, possui ao todo 256 endereços diferentes para serem trabalhados.

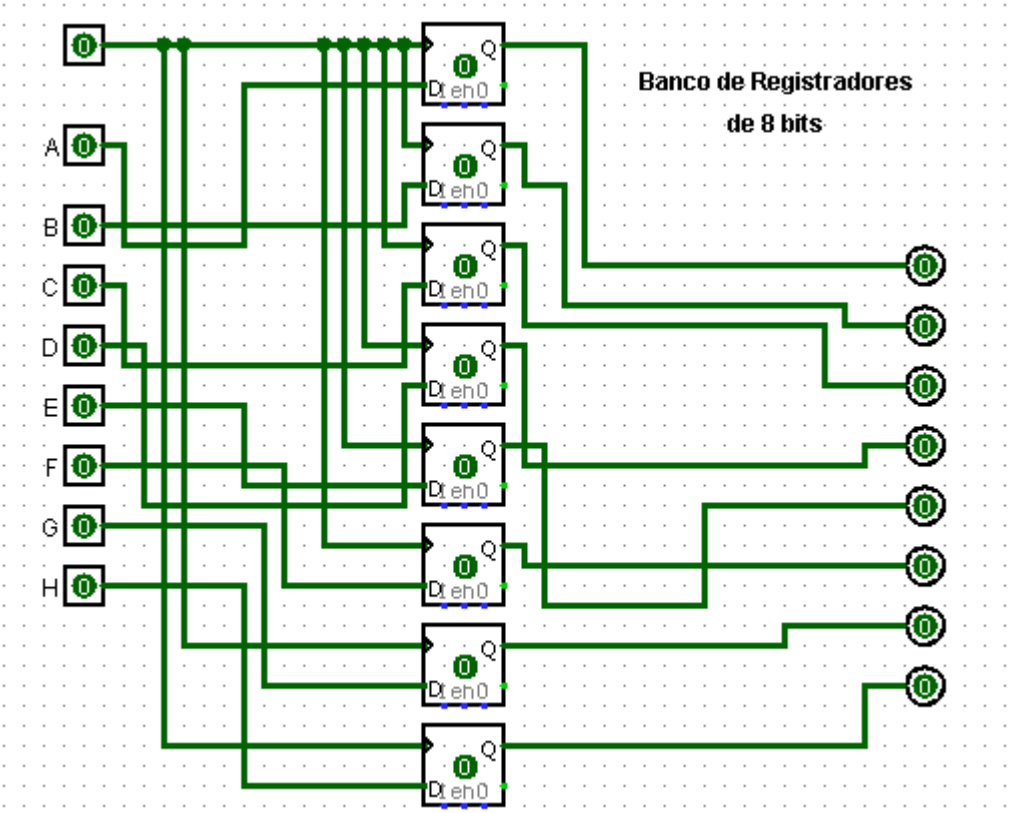
Componente 7: Banco de Registradores de 8 bits.

Descrição pinos, lógica e funcionalidade

Para Banco de Registradores, a arquitetura de circuito é de certa forma, bem simples e intuitiva, utilizando apenas 3 tipos de pinos/componentes, que são eles:

- 8x pinos de entrada de dados: Esses pinos serão os responsáveis por armazenar os dados recebidos para o circuito.
- 8x registradores: Os registradores serão responsáveis por armazenar os dados de entrada de cada pino até que o circuito receba um impulso de clock para seguir a atividades.
- 8x pinos de saída de dados: Os pinos de saída de dados exibirão no circuito durante o do impulso de clock qual o valor que foi mantido ou alterado nos registradores a partir da atividade do clock.
- 1x pino de clock: O impulso de clock faz com que os valores armazenados na entrada de dados sejam lidos e interpretados pelo circuito, sendo assim atualizados nos registradores antes de serem apresentados.

Imagem dos componentes do circuito



Resultado de testes

Os resultados da tabela verdade para o circuito integrado de Banco de Registradores de 8 bits apresentou todos os resultados lógicos de maneira eficaz, partindo exatamente do princípio de funcionamento do circuito, mostrando os atuais dados de entrada para o circuito e quais são os dados de saída após serem armazenados nos registradores e impulsionados por um tick (ciclo de clock). Segue um trecho das primeiras 8 linhas da tabela verdade referente ao circuito:

A	B	C	D	E	F	G	H	x	y	z	u	v	w	s	t
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Descrição dos testes

Durante a fase de *testes,* toda a tabela verdade referente a este circuito foi atualizada a partir dos ciclos de clocks dados no circuito, mostrando o bom funcionamento das atividades e integridades dos componentes usados.

Componente 8: Somador de 8 bits.

Descrição pinos, lógica e funcionalidade

Para o Somador de 8 Bits foram criados 3 componentes que funcionam em forma de camadas e que em conjunto formam o somador, são eles:

- 1x Meio Somador: Esse componente é bem simples, possui duas entradas (a e b) e duas portas lógicas (XOR e AND), e duas saídas (s e co) para os resultados das somas.

- 1x Somador Completo: Para criação do somador completo foi utilizado dois meio somadores, o criado e uma cópia do mesmo, ligados a uma porta OR. Neste caso, são três entradas (a, b e ci) e as mesmas duas saídas do meio somador.

- 1x Somador 4 Bits: Então é criado o somador de 4 bits a partir de 4 somadores completos. Onde vai haver duas entradas (neste caso, a e b seguidos de um número) para cada somador completo), uma entrada Carry In. Uma saída para cada somador completo. Em cada somador a saída co vai se ligar a entrada ci do próximo somador, como uma cascata. E por último, um Carry Out.

Por fim, para criação do somador de 8 bits são utilizados 2 somadores de 4 bits, onde há um aumento do número de pinos de entrada com duas entradas para cada duas entradas dos somadores de 4 bits. Os somadores de 4 bits se conectam pela saída c4 de um e pela entrada c0 do outro. A saída c8 é a saída adicional do somador de 8 bits.

Imagem dos componentes do circuito

Imagem 1

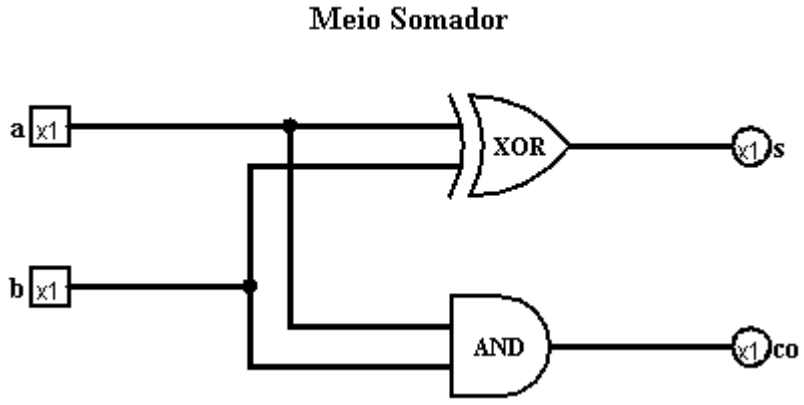
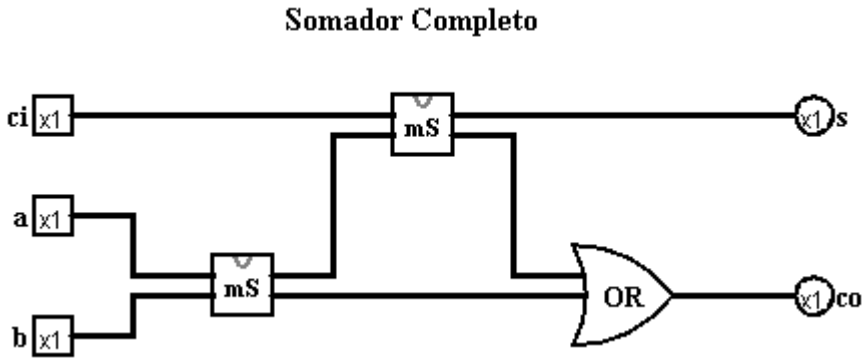
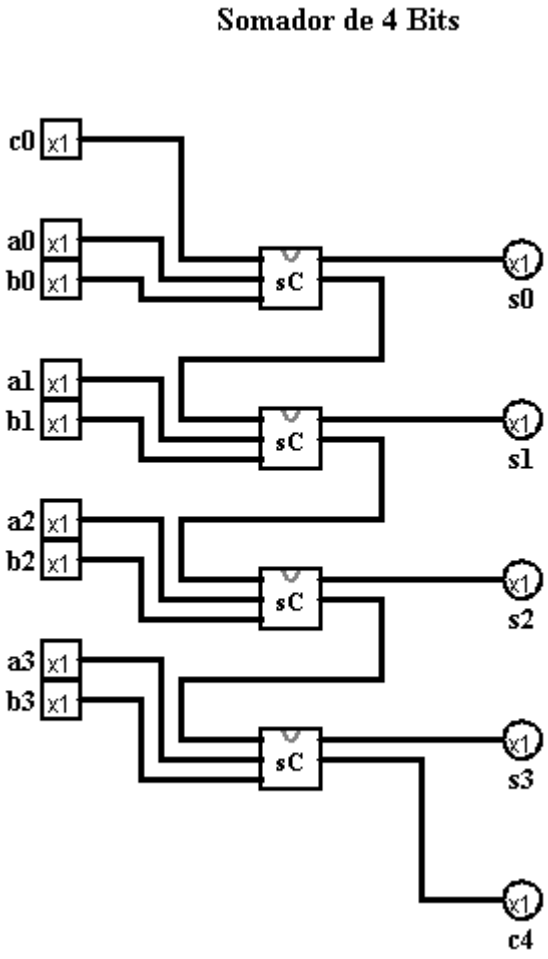


Imagem 2



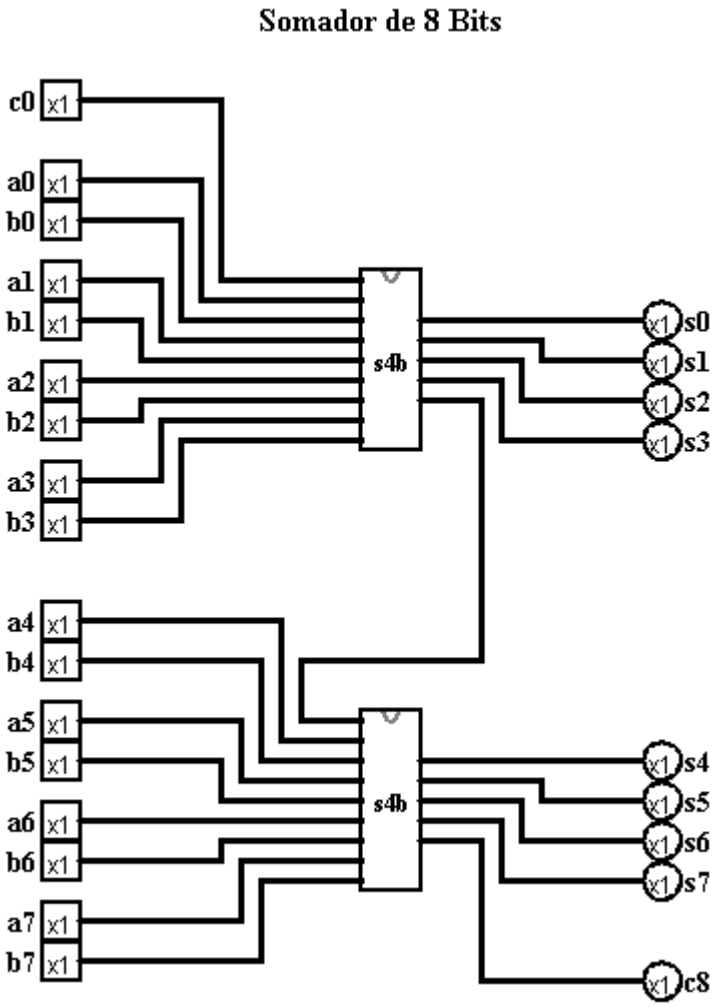
Descrição: mS = Meio Somador

Imagem 3



Descrição: sC = Somador Completo

Imagem 4



Descrição: s4b = Somador de 4 Bits

Resultado de testes

a	b	s	co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabela Verdade – Meio Somador

ci	a	b	s	co
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

Tabela Verdade – Somador Completo

(Ausência das Tabelas Verdade dos Somadores de 4 e 8 Bits devido ao tamanho de ambas)

Descrição dos testes

Durante a fase de testes, todas as tabelas verdade referentes ao Meio Somador, Somador Completo, Somador de 4 Bits e Somador de 8 Bits obtiveram sucesso, foi possível fazer somas em todos os componentes. Foram ligados os pinos de forma que o número das entradas a e b fossem equivalentes aos dois números a serem somados e na saída era mostrado o resultado da soma.

Componente 9: Unidade de controle uni ciclo do MIPS de 16 bits.

Descrição pinos, lógica e funcionalidade
Imagem dos componentes do circuito
Resultado de testes
Descrição dos testes

Componente 10: ULA de 8 bits, utilizando port map, com as seguintes operações: AND, OR, NOT, NOR, NAND, XOR, SHIFT de 2 bits à esquerda, SHIFT de bits à direita, soma e subtração.

Descrição pinos, lógica e funcionalidade

Na imagem a seguir estará exposto como foi construído o port map para ULA de 8 bits agregando as funções que foram mencionadas acima. Esse port map tem em sua estrutura 16 pinos/componentes, nos quais serão todos apresentados e explicados. O funcionamento desse circuito se dá da seguinte maneira: Para cada valor registrado nas entradas de dados A e B, serão realizadas todas as operações dispostas no circuito, de maneira que seus resultados estarão armazenados dentro do multiplexador que da suporte para 8 bits de entrada, sendo todos essas entradas operatrizes, na qual cada operador está setado de maneira a ocupar um bit. Sobre os pinos/componentes do circuito, temos:

- 2x pinos de entrada de dados: Esses pinos, setados como A e B no circuito, receberão os dados solicitados pelo usuário, que passarão a ser os operandos em questão, sendo utilizados em todos os operadores apresentados.
- 1x porta AND: Este é um componente de operação lógico que verificará se ambos os operandos são de valores lógicos do tipo *true* (verdadeiros).
- 1x porta OR: Este é um componente de operação lógico que resulta em um valor lógico do tipo *false* (falso) se, somente se, todos os operandos forem falsos, no caso, A e B.
- 2x portas NOT: Este é um operador do tipo negação, que informa o valor lógico contrário do valor real sobre o operando ao qual estiver conectada, contrariando os resultados que são apresentados na tabela verdade ao fim da execução do circuito.

- 2x Porta NOR: Essas portas são operadores booleanos que resultam na negação do valor operador, portanto, ele apenas tem resultado lógico igual a verdadeiro se ambos os operandos forem falsos.

- 1x porta NAND: Está porta equivale a uma negação de conjunção, conhecida como conectivo da negação alternativa, é verdadeiro se pelo menos um dos operandos do circuito (A ou B) forem falsos.

- 1x porta XOR: Por fim, a última porta do circuito é uma porta de duas entradas que produz na saída o nível lógico 1 quando as entradas forem de valor lógico diferentes, e o nível lógico 0 quando as entradas forem de valores lógicos iguais.

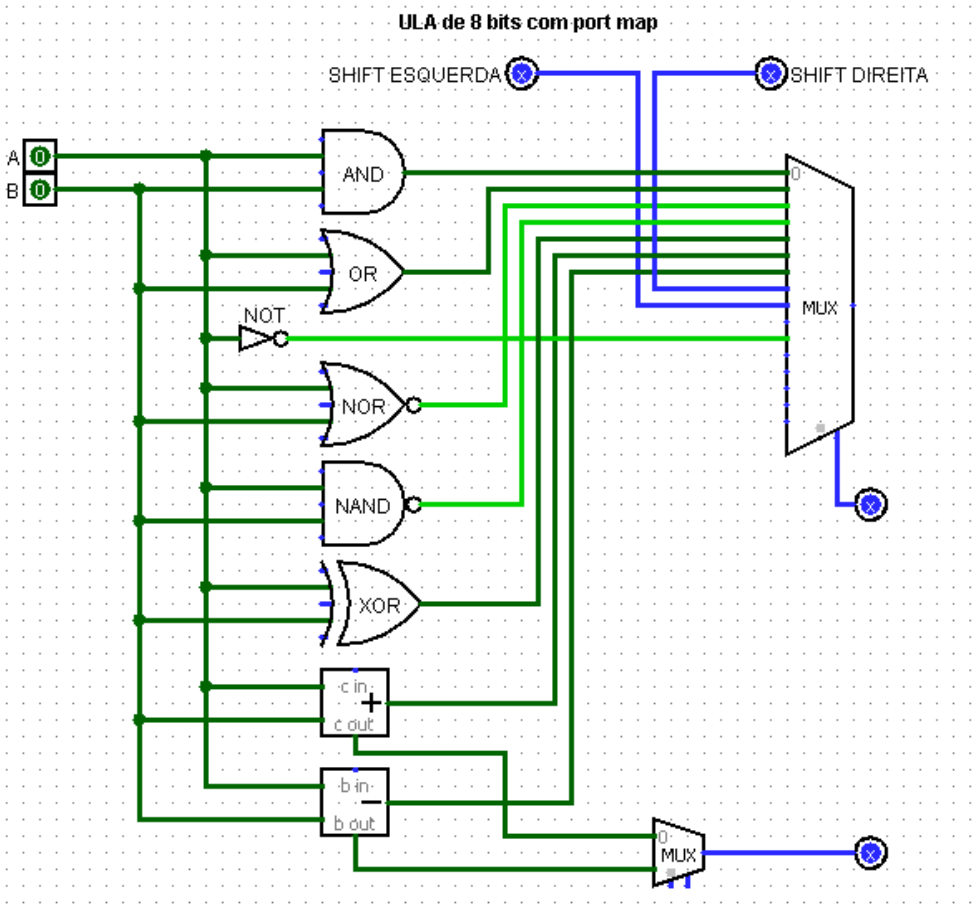
- 1x somador: Esse componente irá realizar a operação aritmética de soma entre os operandos.

- 1x subtração: Esse componente irá realizar a operação aritmética de subtração entre os operandos.

- 2x multiplexadores: Os multiplexadores são responsáveis por armazenar os dados obtidos nas operações e, a partir de cada impulso de clock que for dado no circuito, lê estes dados e passá-los pelas saídas que estiverem ligada ao mesmo.

- 4x pinos de saída de dados: Por último, os pinos responsáveis por apresentar quais são os resultados sobre as operações realizadas em todos as 6 portas e 2 operadores presentes no circuito, relacionadas as atribuições de A e B.

Imagem dos componentes do circuito



Resultado de testes

Não foram realizados testes deste port map. Segue um modelo de como séria a tabela verdade para o esquema do circuito:

A	B	SHIFTESQUERDA	SHIFTDIREITA	x	y
0	0	x	x	x	x
0	1	x	x	x	x
1	0	x	x	x	x
1	1	x	x	x	x

Descrição dos testes

A realização dos testes não foi executada em função do fato que este circuito apresentado para o tópico não trata-se de uma ULA elaborada, mas sim um port map com suas características que receberia os valores resultantes do funcionamento do circuito assim que toda a ULA executasse as etapas do processo após um ciclo de clock.

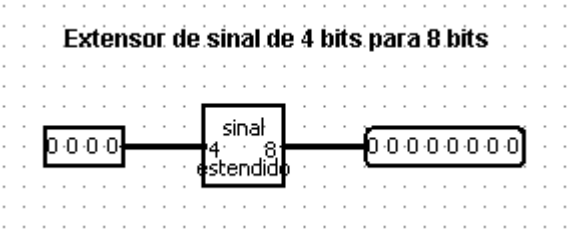
Componente 11: Extensor de sinal de 4 bits para 8 bits.

Descrição pinos, lógica e funcionalidade

Para a produção do circuito de extensor de sinais foram utilizados 3 componentes básicos na estrutura, onde os dados são primeiramente inseridos, depois analisados e por fim, transcrevidos e traduzidos de acordo com a proposta do circuito. Os pinos/componentes utilizados foram:

- 1x pino de entrada de dados: Esse componente armazenara a sequência de códigos binários que o usuário registrar, dispostos em 4 bits.
- 1x extensor de bits (tipo sinal): Este irá interpretar o sinal do código disposto em 4 bits, e ao interpretar como o seu sinal é emitido, ele fará a extensão deste sinal e transcrição para um código sequencial binário disposto em 8 bits.
- 1x pino de saída de dados: Por fim, este pino irá apresentar ao usuário o resultado da conversão de sinal realizado pelo circuito, mostrando apenas o código binário de 8 bits como apresentado na imagem a seguir.

Imagem dos componentes do circuito



Resultado de testes

Teste 1

Teste 2

Teste 3

Descrição dos testes

Os testes ocorreram com *êxito*, como apresentado nas *imagens acima*, pois a cada código sequencial em binários dispostos em 4 bits é atualizado o sinal para os códigos sequenciais dispostos em 8 bits, fazendo assim a atualização e conversão dos dados.

Componente 12: Implemente a máquina de estados ao lado.

Descrição pinos, lógica e funcionalidade
Imagem dos componentes do circuito
Resultado de testes
Descrição dos testes

Componente 13: Contador Síncrono.

Descrição pinos, lógica e funcionalidade
Imagem dos componentes do circuito
Resultado de testes
Descrição dos testes

Referências

1 - http://www.cburch.com/logisim/pt/index.html

2 - HENESSY, J. L. ; PATTERSON, D. A. Organização e Projeto de Computadores. Rio de Janeiro: TC 2005.