



FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais

Aula 10

Prof. Héctor Pettenghi

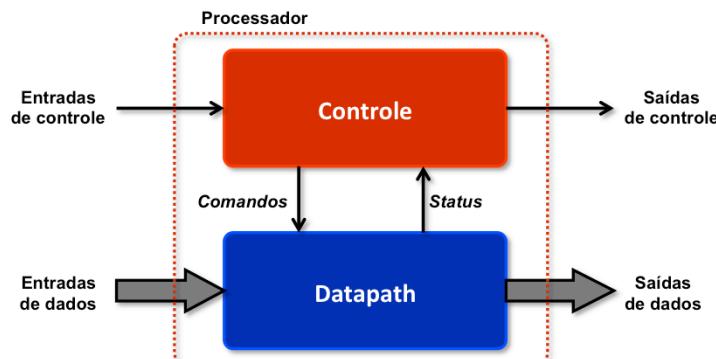
hector@eel.ufsc.br

<http://hectorpettenghi.paginas.ufsc.br>

Material desenvolvido com apoio de arquivos de apresentação do livro de Frank Vahid

RTL Design

- Na aula de hoje:
 - Como associar um datapath a um bloco de controle para construir um **processadores completos mais complexos**.
 - Completar o **RTL (register-transfer level) design**



Nesta aula finalmente veremos como conectar um datapath ao bloco de controle e, assim, completar o RTL design.

RTL Design: Exemplo 1

- **Exemplo:** máq. de venda de refrigerante
 - Supondo que tal máquina possui:
 - Interface com o usuário com saída **s** de 8 bits que indica custo do refrigerante escolhido
 - Mecanismo de recepção de moedas com:
 - saída **c** de 1 bit que indica que uma moeda foi depositada
 - saída **a** de 8 bits que indica o valor da moeda
 - Mecanismo de entrega de refrigerante com entrada **d** de 1 bit que aciona a entrega

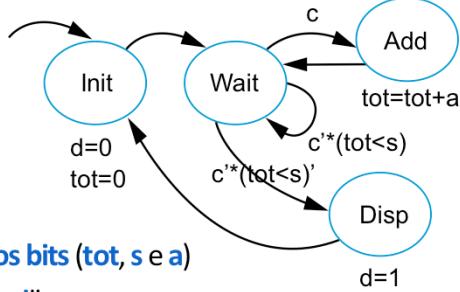


Vejamos este exemplo. Extraímos dele que a partir da interface com o usuário e o mecanismo de recepção de moedas faremos o mecanismo do entrega. Então teremos s, c e a como entradas do nosso circuito e d como saída.

RTL Design: Exemplo 1

- **Exemplo:** máquina de venda de refrigerante

- **Diagrama** de estados:



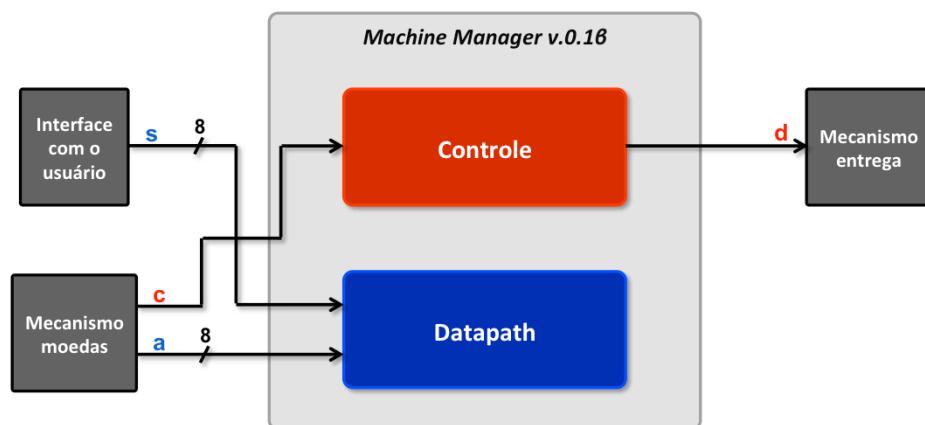
- Tal **diagrama** contém:

- Variáveis com **múltiplos bits** (**tot**, **s** e **a**)
 - Operações de “**alto nível**”:
 - Soma de valores com múltiplos bits (**tot+a**)
 - Atribuição (registro) de valores (**tot=0**, **tot=tot+a**)
 - Comparação de grandezas (**tot<s**)

- Tem-se então um **diagrama de estados de alto nível**

O diagrama de estados de alto nível para o circuito será o deste slide. Observe as variáveis com múltiplos bits e as operações de alto nível necessárias.

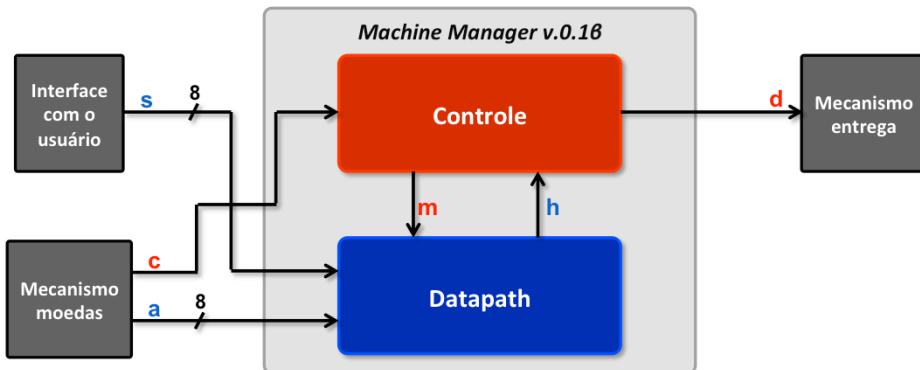
RTL Design: Exemplo 1



- Entradas/saídas de controle: **c** e **d**
- Entradas/saídas de dados: **s** e **a**

O esquema básico do nosso circuito será este. Temos como entrada de controle c, saída de controle d e entradas de dados s e a.

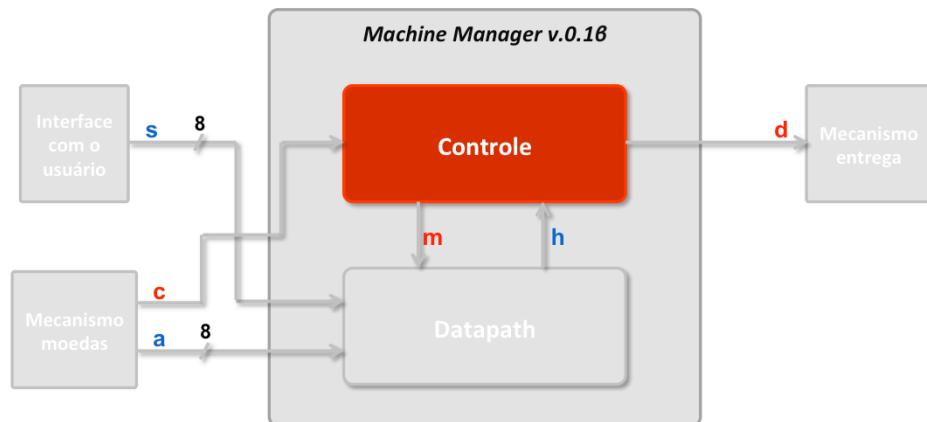
RTL Design: Exemplo 1



- Possíveis sinais de **comando** e **status**:
 - **m** (manda **datapath** somar valor da nova moeda)
 - **h** (informa **controle** que valor depositado é maior que preço do refrigerante escolhido).

Teremos ainda o sinal de controle **m**, que manda o datapath somar o valor da nova moeda, e **h**, que informa o bloco de controle se o valor do refrigerante foi atingido ou ultrapassado. Note que caso ultrapasse o valor, não ficam créditos nem há troco da máquina.

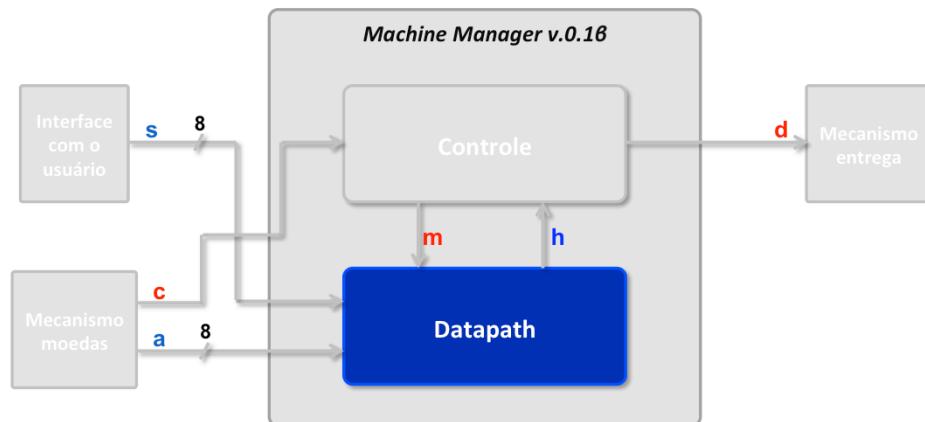
RTL Design: Exemplo 1



- Projeto do **Bloco de Controle**: usando a técnica descrita nesta aula

O bloco de controle será desenvolvido de acordo com a técnica descrita nesta aula.

RTL Design: Exemplo 1

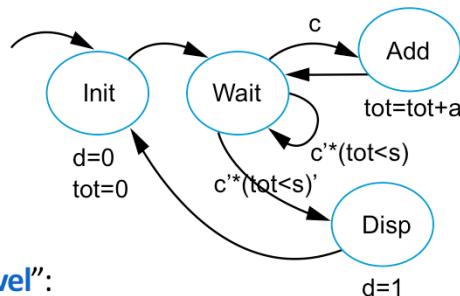


- **Datapath**: Realiza operações com **entradas/saídas de dados**

Mas começemos pelo datapath.

RTL Design: Exemplo 1 (Datapath)

- Para a máquina de venda de refrigerante:
 - Diagrama de estados de alto nível:

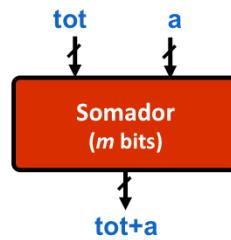


- Operações de “alto nível”:
 - Soma de valores com múltiplos bits ($tot+a$)
 - Atribuição (registro) de valores ($tot=0$, $tot=tot+a$)
 - Comparação de grandezas ($tot < s$)

Voltando às operações de alto nível separadas anteriormente, vejamos quais são os componentes necessários para o datapath.

RTL Design: Exemplo 1 (Datapath)

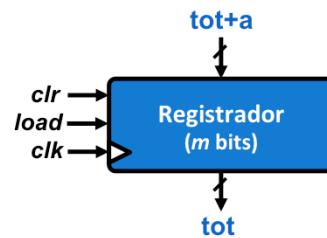
- Na **máquina de venda de refrigerante**:
 - Soma de valores com múltiplos bits (**tot+a**)
 - Somador da **Aula 5**



Precisamos de um somador.

RTL Design: Exemplo 1 (Datapath)

- Na **máquina de venda de refrigerante**:
 - Atribuição (registro) de valores ($\text{tot}=0$, $\text{tot}=\text{tot}+a$)
 - Registrador com FFs Tipo D com *load* e *clear* das **Aulas 7, 8 e Lab**

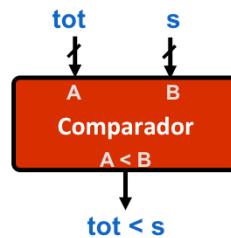


$$\begin{aligned}\textcolor{red}{load = 1} &\rightarrow \text{tot} = \text{tot} + a \\ \textcolor{red}{clr = 1} &\rightarrow \text{tot} = 0\end{aligned}$$

De um registrador para tot, com flip-flops do tipo d, load e clear.

RTL Design: Exemplo 1 (Datapath)

- Na **máquina de venda de refrigerante**:
 - Comparação de grandezas (**tot < s**)
 - Comparador, o qual pode ser projetado usando **Aula 3**

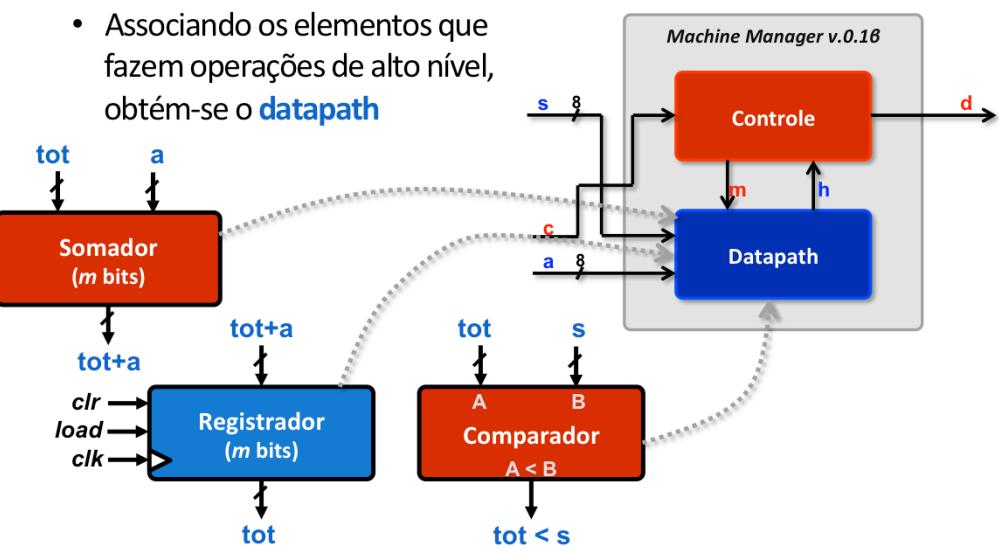


Precisamos também de um comparador de grandezas. Lembrando que todos estes componentes foram vistos na aula passada.

RTL Design: Exemplo 1 (Datapath)

- Na **máquina de venda de refrigerante**:

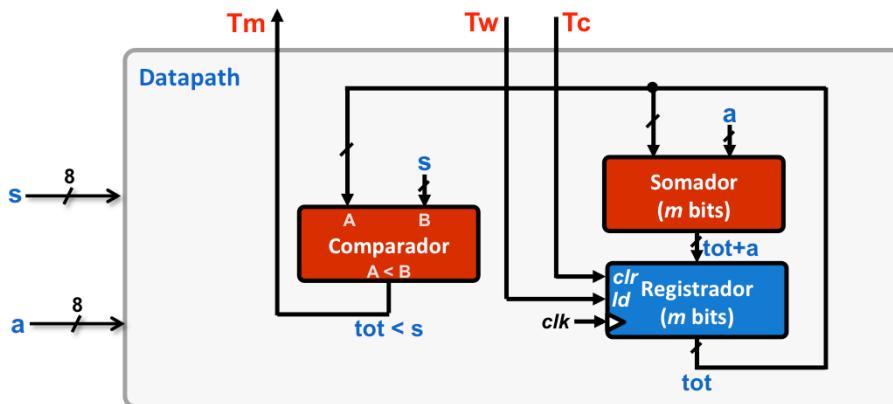
- Associando os elementos que fazem operações de alto nível, obtém-se o **datapath**



Associando os elementos que fazem as operações de alto nível. Obtemos o datapath.

RTL Design: Exemplo 1 (Datapath)

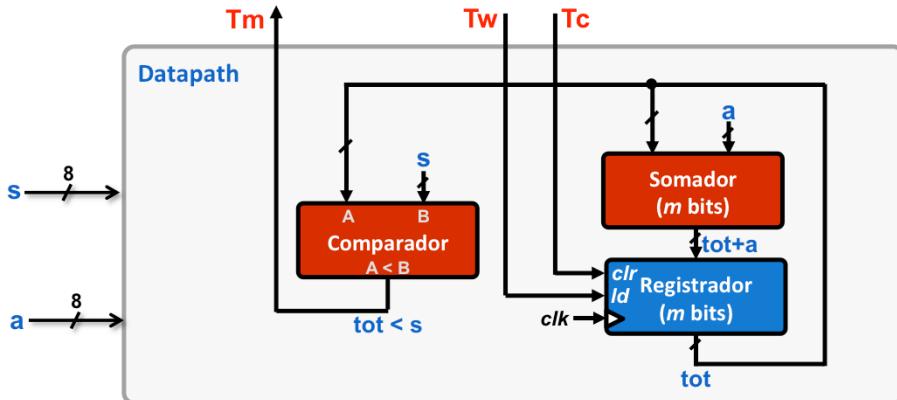
- Na **máquina de venda de refrigerante**:
 - Associando os elementos que fazem operações de alto nível, obtém-se o **datapath**



Tais são as conexões entre os elementos. Veja que este circuito é bem semelhante ao contador de 100 da aula passada.

RTL Design: Exemplo 1 (Datapath)

- Na máquina de venda de refrigerante:

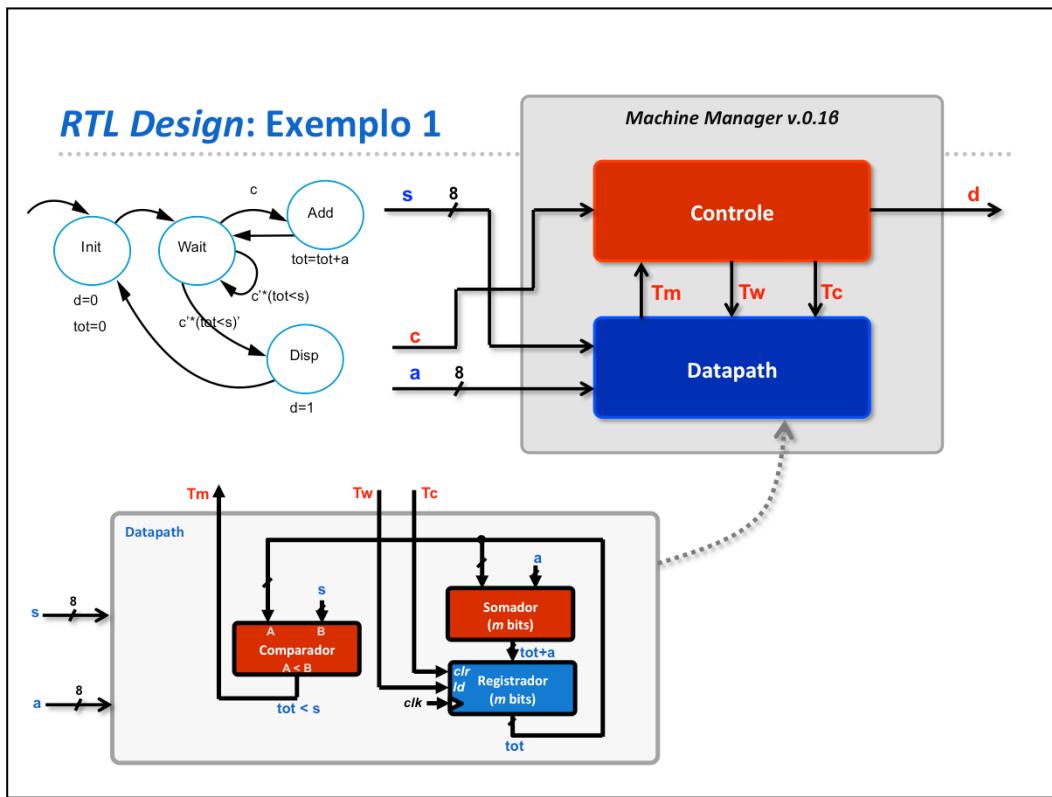


T_m : resultado de $tot < s$

$T_w = 1 \rightarrow tot = tot + a$

$T_c = 1 \rightarrow tot = 0$

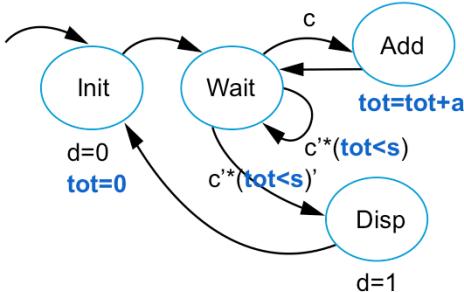
Teremos assim T_m como sinal de "status" e T_w e T_c como "comandos". Veja que T_m é o resultado $tot < s$, enquanto quando $T_w=1$, $tot=tot+a$, e quando $T_c=1$, $tot=0$.



Com os "comandos" e o sinal de "status" definidos podemos transformar o diagrama de estados de alto nível em uma FSM que possa compor o bloco de controle.

RTL Design: Exemplo 1

- Na máq. de venda de refrigerante.
 - Primeiro passo: **diagrama de estados de alto nível**

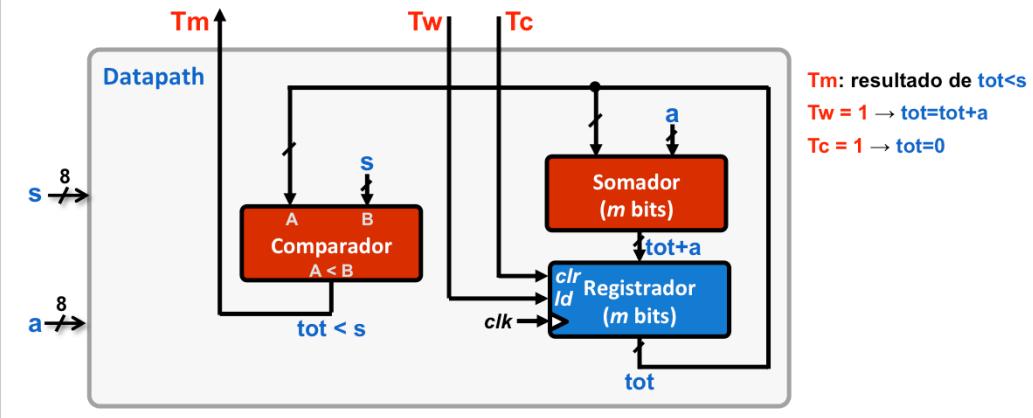


- Operações de “**alto nível**”:
 - Soma de valores com múltiplos bits (**tot+a**)
 - Atribuição (registro) de valores (**tot=0**, **tot=tot+a**)
 - Comparação de grandezas (**tot<s**)

Recapitulando: primeiro definimos o diagrama de estados de alto nível para o problema.

RTL Design: Exemplo 1

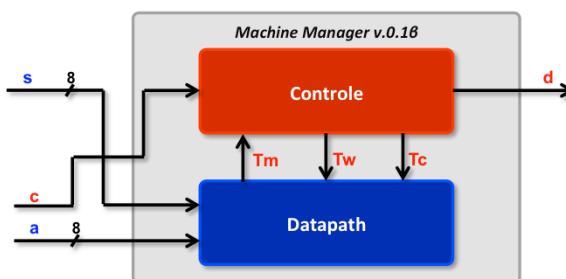
- Na máq. de venda de refrigerante.
 - Primeiro passo: **diagrama de estados de alto nível**
 - Segundo passo: criar **datapath**
(permite converter **diag. de alto nível** em **FSM**)



Depois, criamos definimos os elementos do datapath e o construímos. Tudo isso já havia sido feito na aula passada.

RTL Design: Exemplo 1

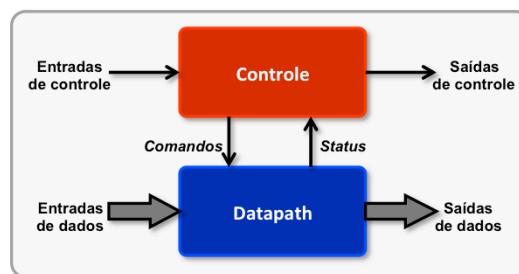
- Na máq. de venda de refrigerante
 - **Primeiro passo:** **diagrama de estados de alto nível**
 - **Segundo passo:** criar **datapath**
(permite converter **diag. de alto nível** em **FSM**)
 - **Terceiro passo:** conectar **datapath** ao **bloco de controle**
 - **Quarto passo:** projetar **bloco de controle**



A seguir, vamos descobrir como conectar o datapath ao bloco de controle e, por último, como projetar este último.

RTL Design: Exemplo 1

- Resumo de *RTL Design*
 - **Primeiro passo:** descrever o comportamento geral
 - Usando uma **FSM** de alto nível
 - **Segundo passo:** criar um **datapath**
 - Para converter **FSM** de alto nível em **FSM convencional**
 - **Terceiro passo:** conectar o **datapath** ao **bloco de controle**
 - **Quarto passo:** projetar o **bloco de controle**

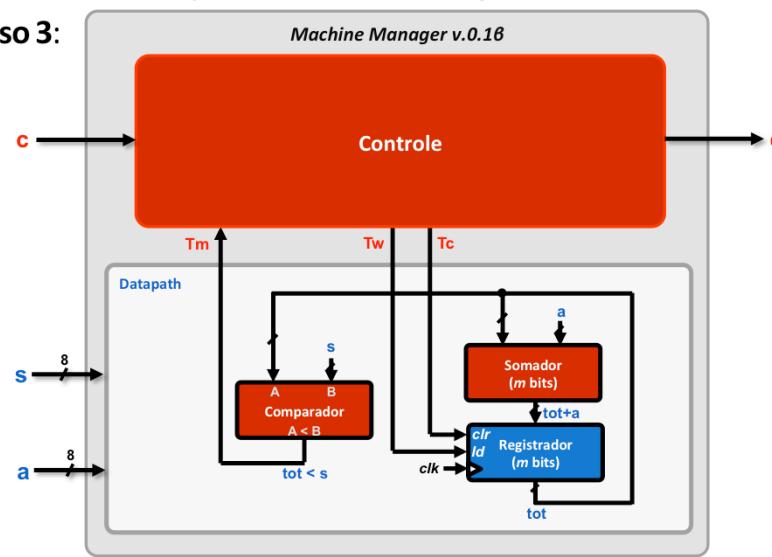


Assim, temos um passo a passo resumido de como realizar o RTL design, que resultará no diagrama geral apresentado neste slide.

RTL Design: Exemplo 1

- Continuando: máq. de venda de refrigerante

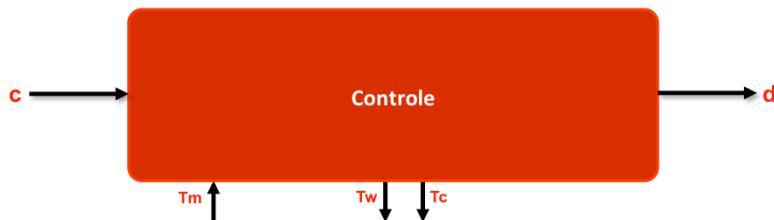
- Passo 3:



Continuando...

RTL Design: Exemplo 1

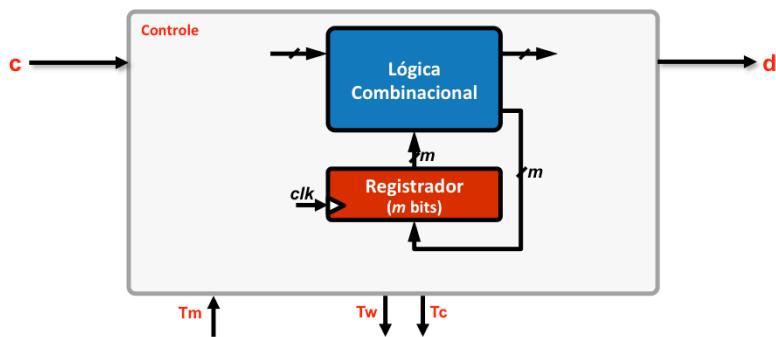
- **Continuando:** máq. de venda de refrigerante
 - **Passo 3:** conectar o **datapath** ao **bloco de controle**
 - **Passo 4:** projetar o **bloco de controle**
 - Usando **Aulas 7 e 8**



A conexão entre o datapath e bloco de controle é feita através dos sinais de status de controle. Vamos ver como construir o bloco de controle e m si.

RTL Design: Exemplo 1

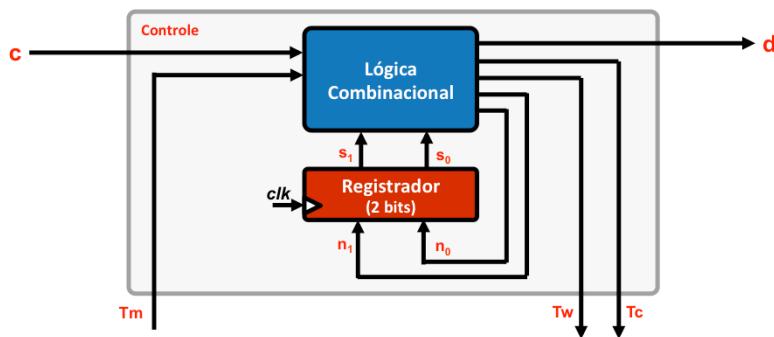
- Continuando: máq. de venda de refrigerante
 - Passo 3: conectar o datapath ao **bloco de controle**
 - Passo 4: projetar o **bloco de controle**
 - Usando **Aulas 7 e 8**



Ele é composto por um bloco de lógica combinacional e um registrador. Ou seja, para construir o bloco de controle basta fazermos um projeto FSM.

RTL Design: Exemplo 1

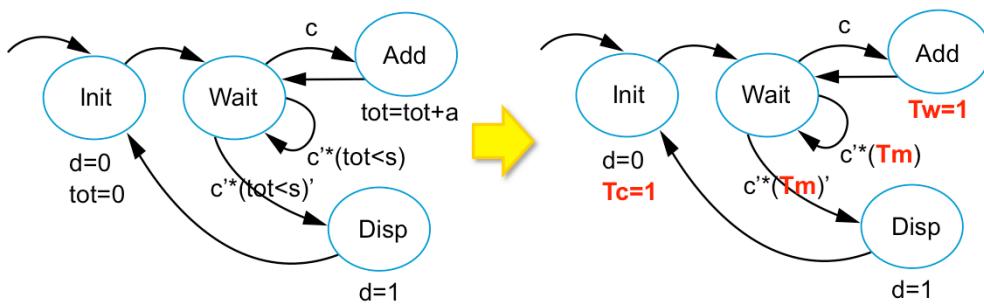
- Continuando: máq. de venda de refrigerante
 - Passo 3: conectar o datapath ao **bloco de controle**
 - Passo 4: projetar o **bloco de controle**
 - Usando Aulas 7 e 8



No nosso exemplo, temos as seguintes conexões entre bloco de controle, registrador e demais entradas e saídas do problema.

RTL Design: Exemplo 1

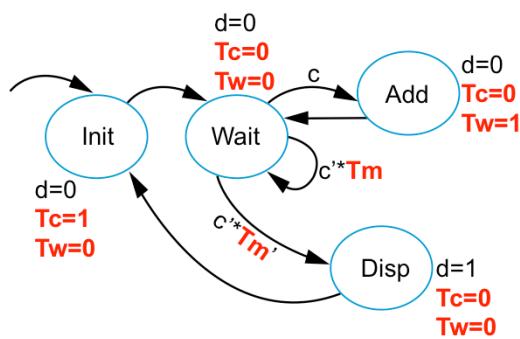
- **Continuando:** máq. de venda de refrigerante
 - **Passo 3:** conectar o **datapath** ao **bloco de controle**
 - **Passo 4:** projetar o **bloco de controle**
 - Usando **Aulas 7 e 8**
 - Convertendo **FSM**:



Para que seja possível fazer este projeto, começamos convertendo o diagrama de estados de alto nível em uma FSM, como apresentado neste slide e no próximo.

RTL Design: Exemplo 1

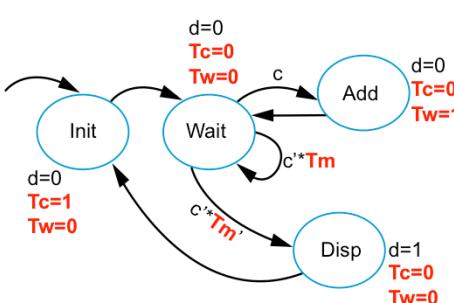
- **Continuando:** máq. de venda de refrigerante
 - **Passo 3:** conectar o **datapath** ao **bloco de controle**
 - **Passo 4:** projetar o **bloco de controle**
 - Usando **Aulas 7 e 8**
 - Convertendo **FSM**:



Este será o nosso bloco de controle.

RTL Design: Exemplo 1

- Projetando o controlador (assunto da Aula 7)

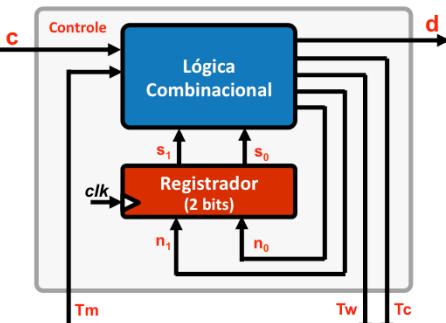


	s1	s0	c	Tm	n1	n0	d	Tw	Tc
Init	0	0	0	0	0	1	0	0	1
	0	0	0	1	0	1	0	0	1
	0	0	1	0	0	1	0	0	1
	0	0	1	1	0	1	0	0	1
Wait	0	1	0	0	1	1	0	0	0
	0	1	0	1	0	1	0	0	0
	0	1	1	0	1	0	0	0	0
	0	1	1	1	1	0	0	0	0
Add	1	0	0	0	0	1	0	1	0
	1	0	0	1	1	0	0
	1	1	0	0	0	0	0	0	0
	1	1	0	1	0	0	0
Disp	1	1	0	0	0	0	1	0	0
	1	1	0	1	0	0	0
	1	1	1	0	0	0	0
	1	1	1	1	0	0	0

A partir desse diagrama de estados obtemos a seguinte tabela de transição de estados. Note que s1, s0, n1 e n0 são variáveis de estado e que Init="00", Wait="01", Add="10" e Disp="11".

RTL Design: Exemplo 1

- Projetando o controlador (assunto da Aula 7)



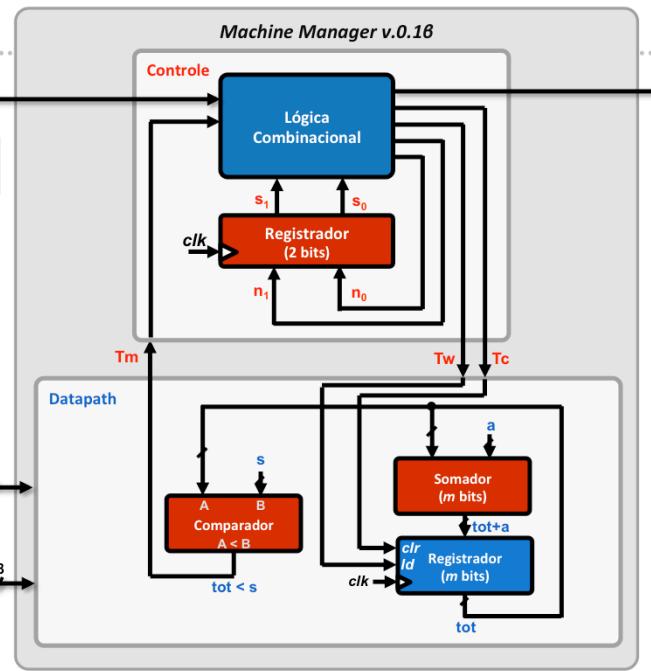
	s1	s0	c	Tm	n1	n0	d	0	1
Init	0	0	0	0	0	1	0	0	1
Wait	0	1	0	0	1	1	0	0	0
Add	1	0	0	0	0	1	0	1	0
Disp	1	1	0	0	0	0	1	0	0
	
	

Por último, basta tirar as equações da tabela verdade e inserí-las no bloco de lógica combinacional.

RTL Design

- Final:

	s1	s0	c	Ξ	n1	n0	d	Tm	Tc
Init	0	0	0	0	0	1	0	0	1
	0	0	0	1	0	1	0	0	1
	0	0	1	0	0	1	0	0	1
	0	0	1	1	0	1	0	0	1
Wait	0	1	0	0	1	1	0	0	0
	0	1	0	1	0	1	0	0	0
	0	1	1	0	1	0	0	0	0
	0	1	1	1	1	0	0	0	0
Add	1	0	0	0	0	1	0	1	0
				
Disp	1	1	0	0	0	0	1	0	0
				



Aqui está a solução final com a tabela de transição de estados para a lógica combinacional do controle.

Exemplo 2: Multiplicador Sequencial

- Projetar um circuito que faça **multiplicação**
 - Duas entradas de **4 bits** representadas por **A** e **B**
 - Multiplicação deve ser feita acumulando **A** por **B** vezes
 - Entrada **s** indica que a multiplicação deve ser iniciada
 - Saída **b** deve ficar em nível alto enquanto o circuito estiver calculando a multiplicação
 - Resultado deve ser mostrado em saída **R** de **oito bits** somente ao final do processo

Vamos ao segundo exemplo: um multiplicador sequencial.

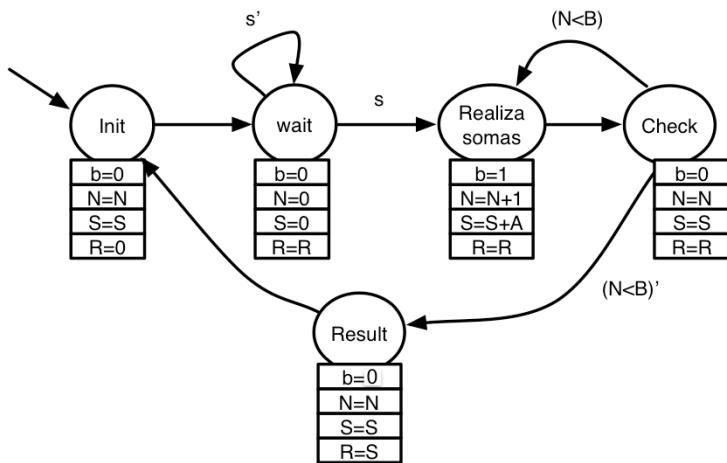
Exemplo 2: Multiplicador Sequencial

- Passo 1: **FSM de alto nível**
- Passo 2: **Datapath** (bloco operacional)
- Passo 3: Conectar **Datapath** e **Bloco de Controle**
- Passo 4: Projetar **Bloco de Controle**

Lembrando dos passos do RTL design.

Exemplo 2: Multiplicador Sequencial

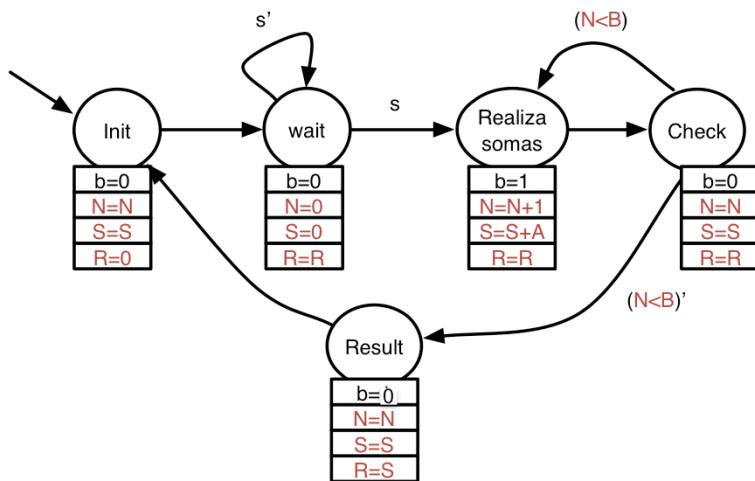
- Passo 1: **FSM de alto nível**



Começamos montando o diagrama de estados de alto nível. Observe que devemos usar, no final, $R=S-A$ para possibilitar a multiplicação por 0.

Exemplo 2: Multiplicador Sequencial

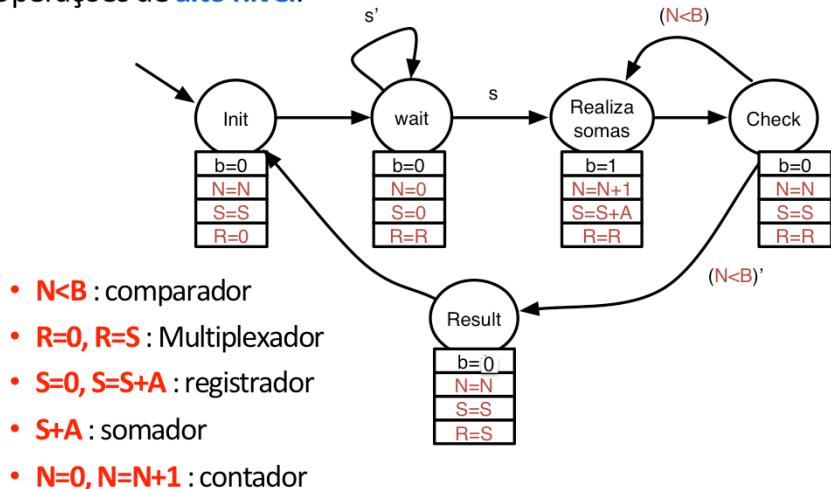
- Passo 2: **Datapath** (bloco operacional)
 - Operações de **alto nível**:



Para o passo 2, note as operações de alto nível destacadas em vermelho ($R=0$, $R=S-A$, $S=0$, $S=S+A$, $N=0$, $N=N+1$, $N \leq B$).

Exemplo 2: Multiplicador Sequencial

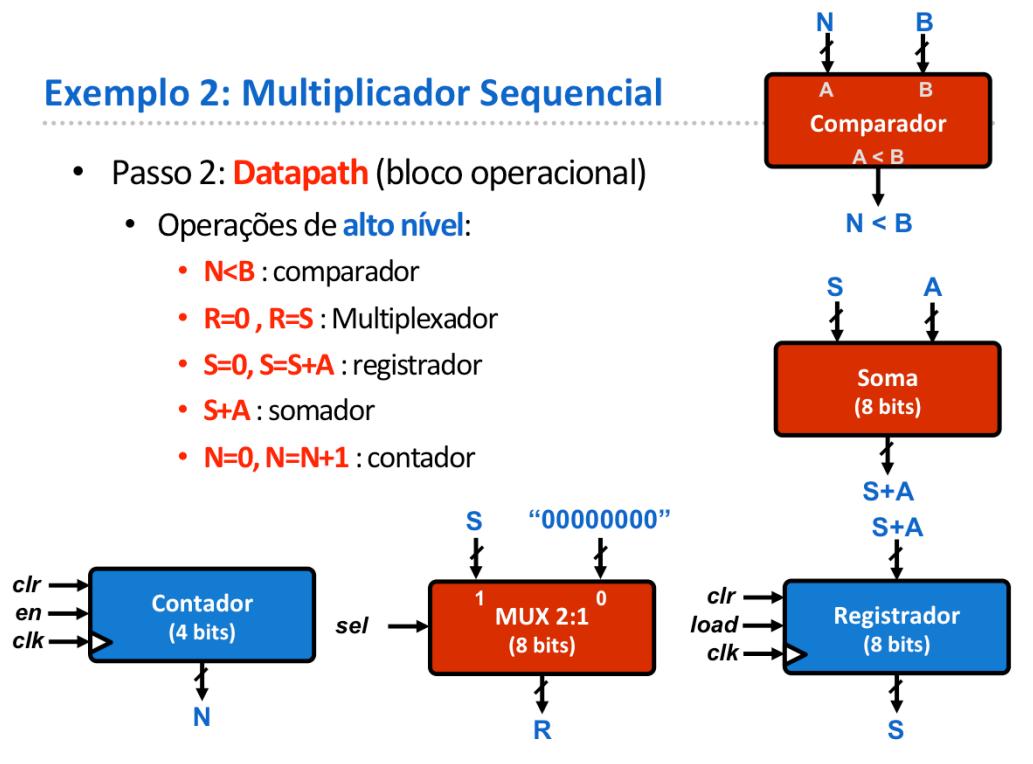
- Passo 2: **Datapath** (bloco operacional)
 - Operações de **alto nível**:



A partir disso sabemos que precisaremos destes componentes no datapath.

Exemplo 2: Multiplicador Sequencial

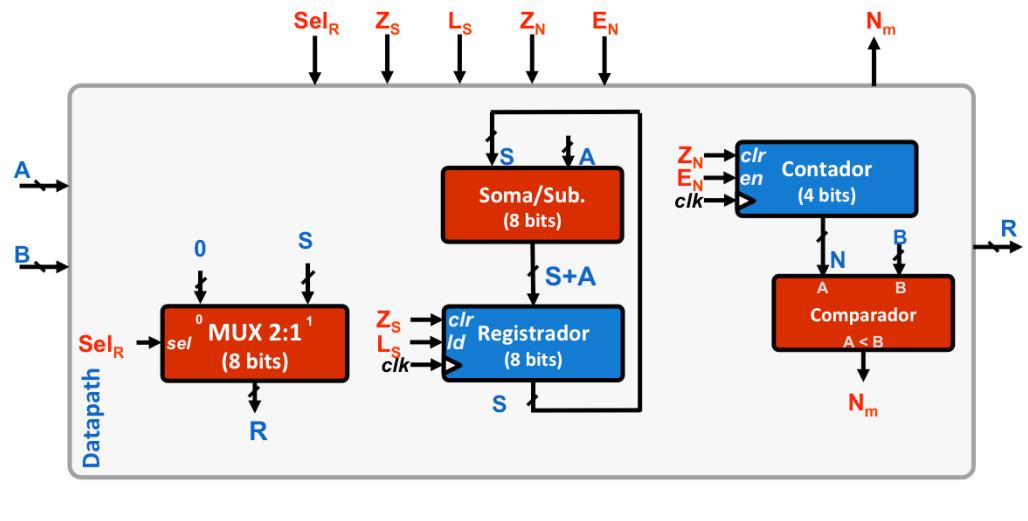
- Passo 2: **Datapath** (bloco operacional)
 - Operações de **alto nível**:
 - N<B** : comparador
 - R=0 , R=S** : Multiplexador
 - S=0, S=S+A** : registrador
 - S+A** : somador
 - N=0, N=N+1** : contador



Eis os componentes.

Exemplo 2: Multiplicador Sequencial

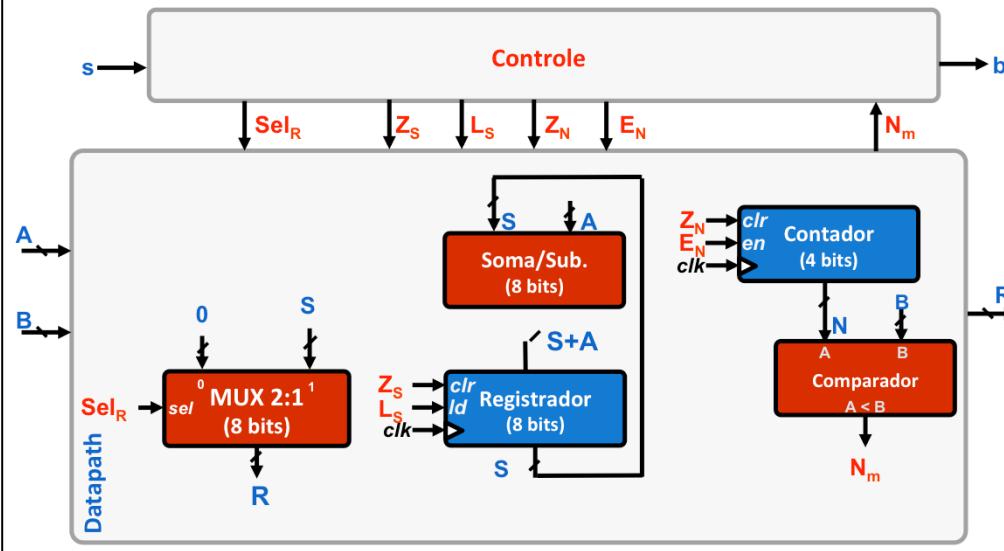
- Passo 2: **Datapath** (bloco operacional)



Estas serão as conexões entre os componentes e seus sinais de controle e de status.

Exemplo 2: Multiplicador Sequencial

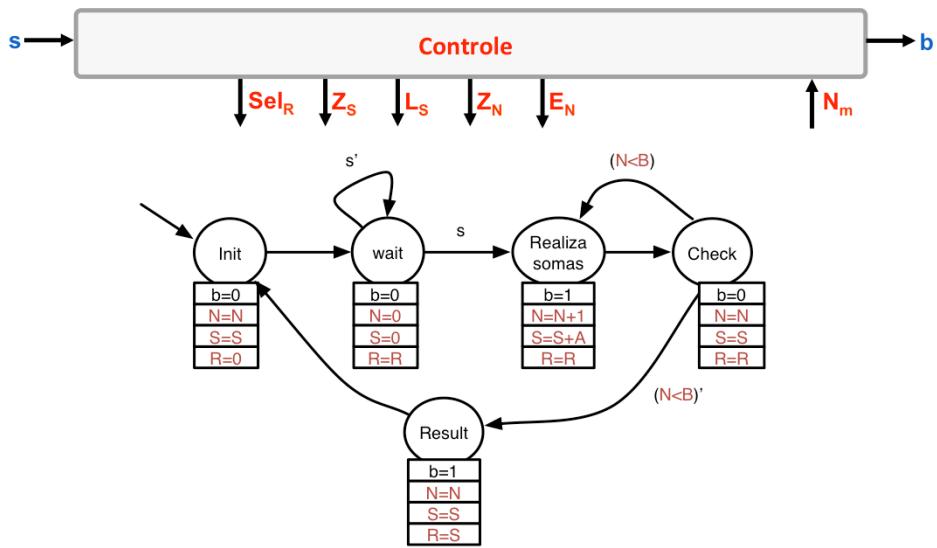
- Passo 3: Conectar Datapath e Bloco de Controle



A partir disso realizamos o passo 3, conectar o datapath ao bloco de controle.

Exemplo 2: Multiplicador Sequencial

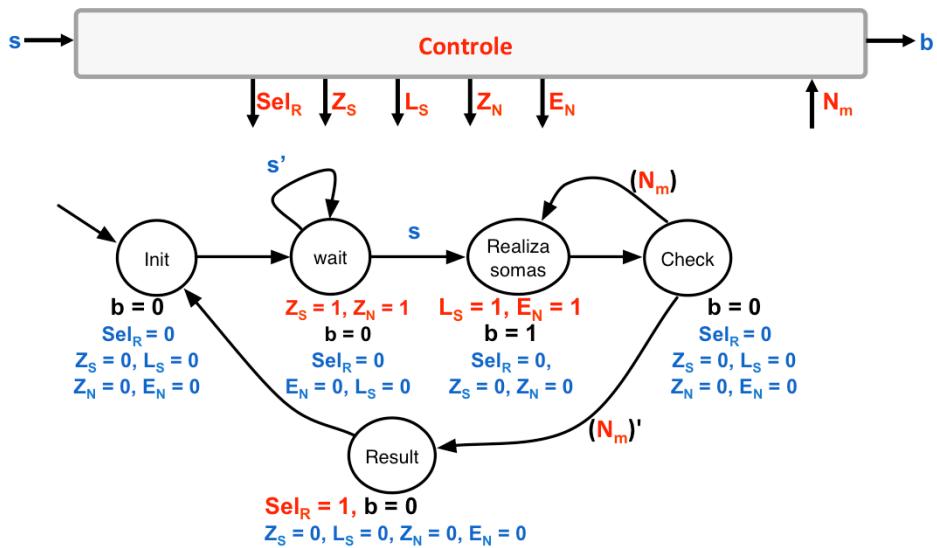
- Passo 4: Projetar Bloco de Controle



Por último, projetamos o bloco de controle. Lembrando das operações de alto nível...

Exemplo 2: Multiplicador Sequencial

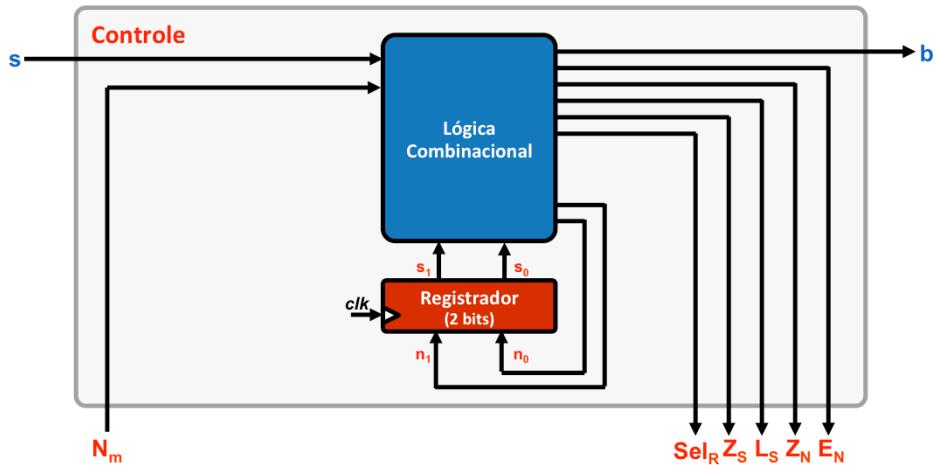
- Passo 4: Projetar Bloco de Controle



...transformamos a FSM de alto nível em uma FSM de baixo nível.

Exemplo 2: Multiplicador Sequencial

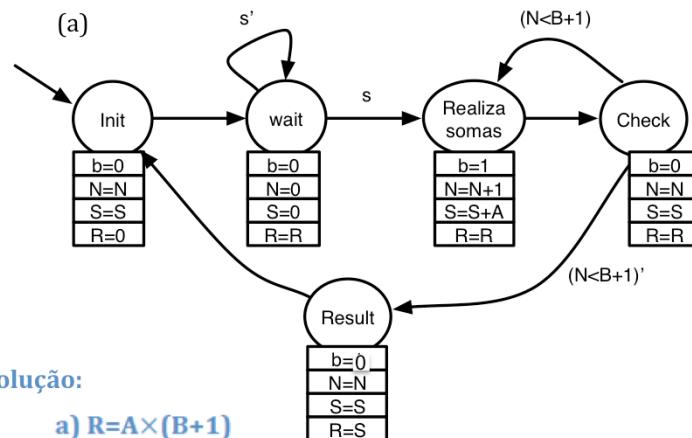
- Passo 4: Projetar **Bloco de Controle**



A partir disso, basta seguir as etapas de construção de uma FSM e teremos o multiplicador sequencial.

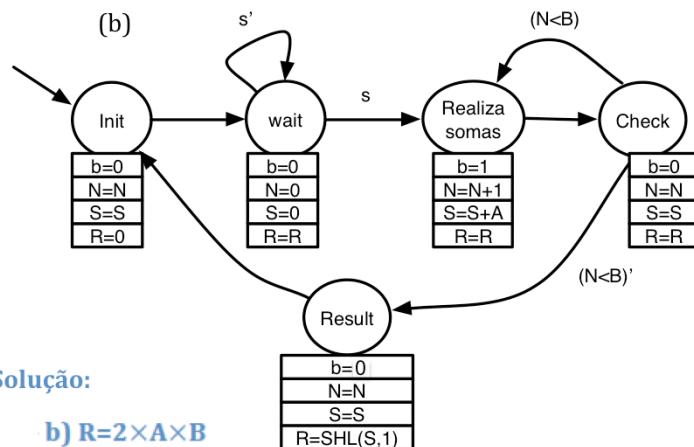
PROBLEMAS

- **Problema 10.1** Qual operação esta a ser mostrada na saída R no estado “Mostra resultado”?



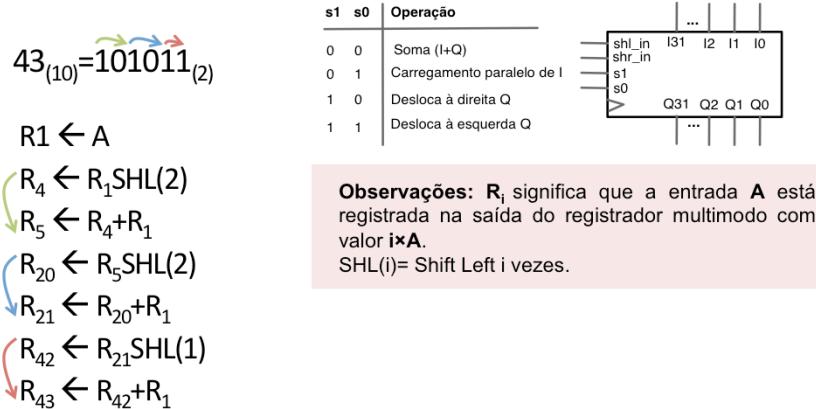
PROBLEMAS

- **Problema 10.1** Qual operação esta a ser mostrada na saída R no estado “Mostra resultado”?



Exemplo 3: Multiplicador por constante Sequencial

- Faça a FSM do controlador (máquina de baixo nível) para a multiplicação pela constante 43 usando o registrador multimodo da Figura no *Datapath*. Considere a entrada A de 32 bits.

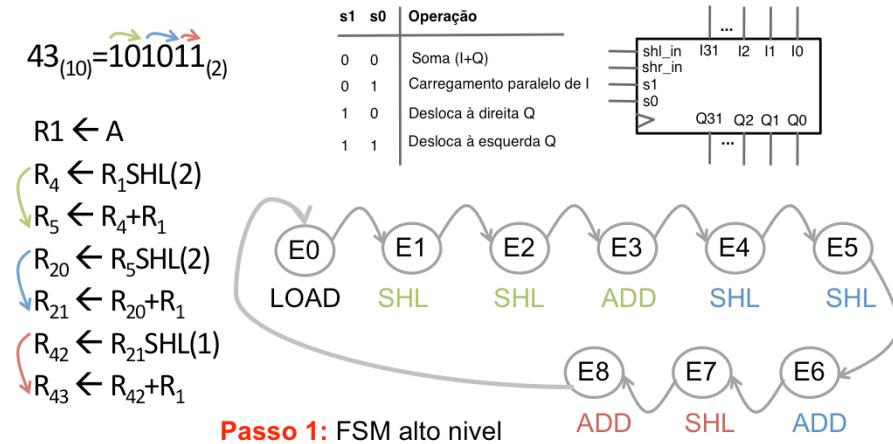


O último exemplo é um multiplicador por constante sequencial, neste caso 43 . Aqui pede-se apenas para fazer a FSM de baixo nível.

Neste slide já temos os passos que serão utilizados para multiplicar um número qualquer A por 43, começando pela carga paralela de A.

Exemplo 3: Multiplicador por constante Sequencial

- Faça a FSM do controlador (maquina de baixo nível) para a multiplicação pela constante 43 usando o registrador multimodo da Figura no *Datapath*. Considere a entrada A de 32 bits.



Isso gera a seguinte máquina de estados de alto nível.

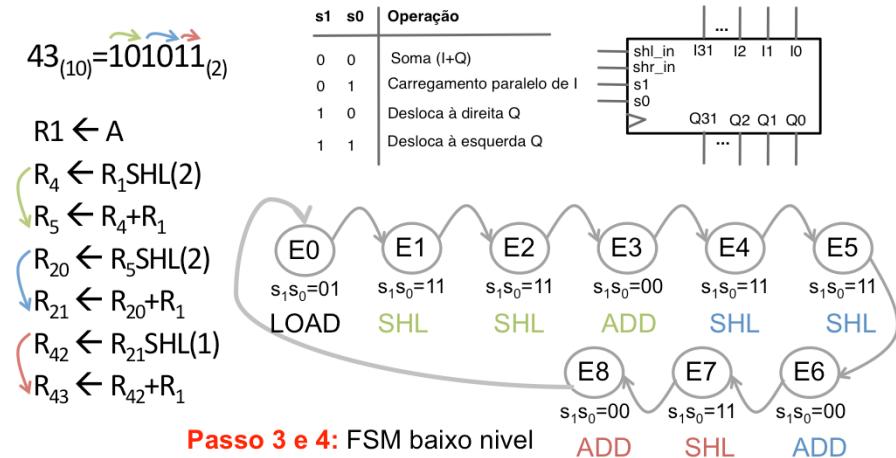
Exemplo 3: Multiplicador por constante Sequencial

- Passo 1: **FSM de alto nível** (No Slide anterior)
- Passo 2: **Datapath** (Registrador Multimodo)
- Passo 3: Conectar **Datapath** e **Bloco de Controle**
- Passo 4: Projetar **Bloco de Controle**

Assim, temos completos os passos 1 e 2, já que o datapath foi dado pelo problema. Vamos ao objetivo do problema: obter a FSM de baixo nível.

Exemplo 3: Multiplicador por constante Sequencial

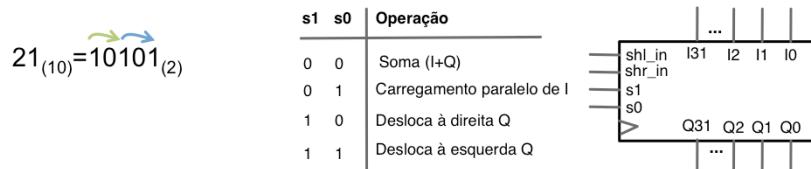
- Faça a FSM do controlador (maquina de baixo nível) para a multiplicação pela constante 43 usando o registrador multimodo da Figura no *Datapath*. Considere a entrada A de 32 bits.



Para isso basta olhar a tabela de operações dada e definir *s1* e *s0* para cada estado. Não há status neste exemplo.

PROBLEMAS

Problema 10.2 Faça a FSM do controlador (maquina de baixo nível) para multiplicação pela constante 21 usando o registrador multimodo da Figura no **Datapath (RTL Design)**. Considere a entrada A de 32 bits.



Justifique quais blocos operacionais deveriam ser introduzidos no *Datapath* para que seja mostrado o número de instruções de deslocamento à esquerda. Dita contagem deve ser zerada cada vez que é realizada uma instrução de carregamento paralelo no registrador multimodo.

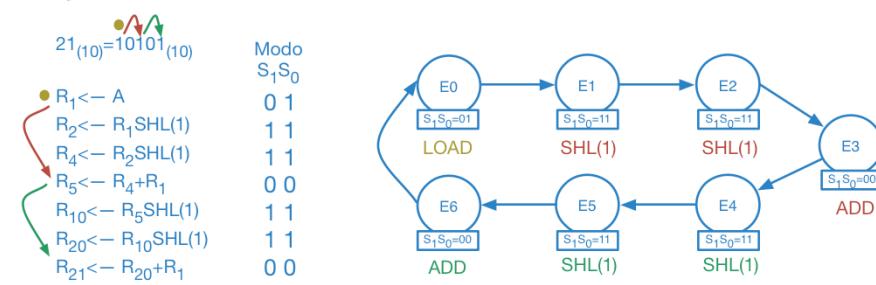
PROBLEMAS

Problema 10.2 Faça a FSM do controlador (maquina de baixo nível) para multiplicação pela constante 21 usando o registrador multimodo da Figura no *Datapath (RTL Design)*. Considere a entrada A de 32 bits.

$$21_{(10)} = 10101_{(2)}$$

s1	s0	Operação
0	0	Soma (I+Q)
0	1	Carregamento paralelo de I
1	0	Desloca à direita Q
1	1	Desloca à esquerda Q

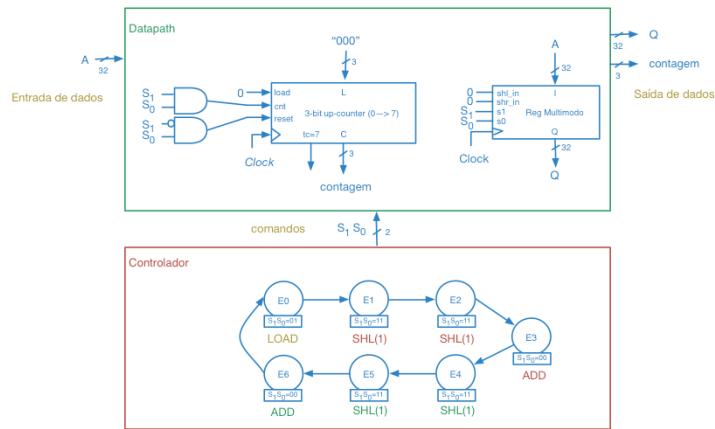
Solução:



PROBLEMAS

Problema 10.2 Justifique quais blocos operacionais deveriam ser introduzidos no *Datapath* para que seja mostrado o número de instruções de deslocamento à esquerda. Dita contagem deve ser zerada cada vez que é realizada uma instrução de carregamento paralelo no registrador multimodo.

Solução:





FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais

Aula 10

Prof. Héctor Pettenghi

hector@eel.ufsc.br

<http://hectorpettenghi.paginas.ufsc.br>

Material desenvolvido com apoio de arquivos de apresentação do livro de Frank Vahid

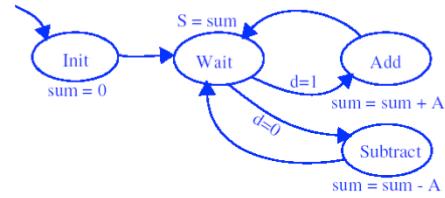
Exercícios

- Exercícios do Livro do **Frank Vahid**
 - **5.1, 5.2, 5.8, 5.10**
 - **5.6 (não cai um desse tamanho na prova, mas ele é interessante para entender todos os passos do projeto RTL)**
- **A versão digital do livro do Frank Vahid está disponível no site da BU**
 - Acesse: <http://www.bu.ufsc.br/design/LivrosEletronicos1.html>
 - Clique no link para “Minha Biblioteca”
http://150.162.1.90/pergamon/biblioteca_s/php/login_usu.php?flag=minhabiblioteca_redirect.php
 - Use sua senha da Biblioteca da UFSC para acessar a “Minha Biblioteca”
 - Procure pelo livro do Vahid...

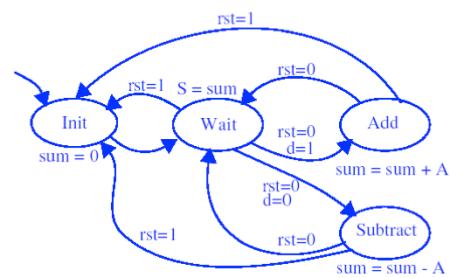
Exercícios

- Resposta do 5.1:

(a)
Inputs: A (8 bits), d (bit)
Outputs: S (32 bits)
Local registers: sum (32 bits)



(b)
Inputs: A (8 bits), d (bit), rst (bit)
Outputs: S (32 bits)
Local registers: sum (32 bits)



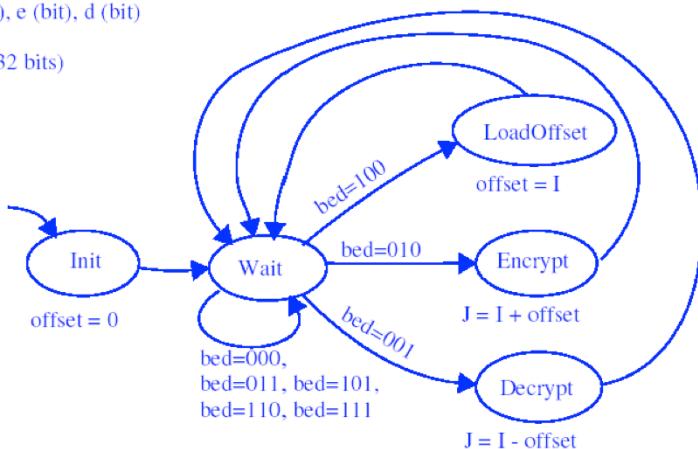
Exercícios

- Resposta do 5.2:

Inputs: I (32 bits), b (bit), e (bit), d (bit)

Outputs: J (32 bits)

Local registers: offset (32 bits)



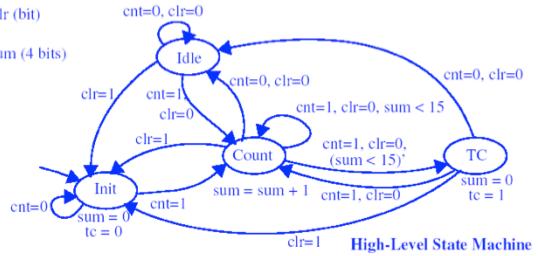
Exercícios

- Resposta do 5.6:

Inputs: cnt (bit), clr (bit)

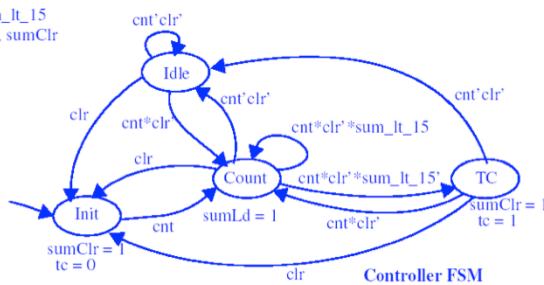
Outputs: tc (bit)

Local registers: sum (4 bits)



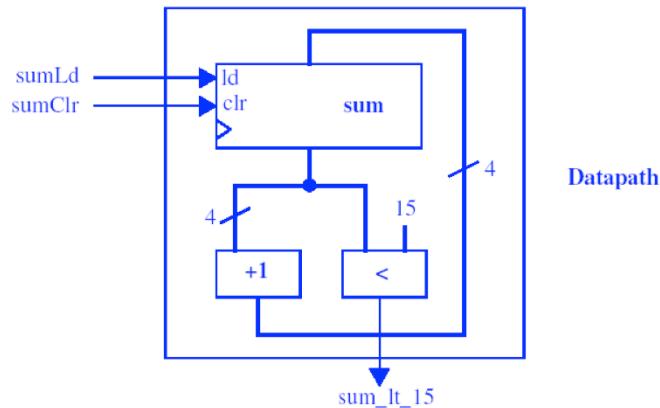
Inputs: cnt, clr, sum_lt_15

Outputs: tc, sumLd, sumClr



Exercícios

- Resposta do 5.6 (continuação):



Exercícios

- Resposta do 5.6 (continuação)

Inputs					Outputs					
	s1	s0	cnt	clr	sum_lt_15	m1	n0	tc	sumId	sumClr
Init	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	1	0	0	0	0	1
	0	0	0	1	0	0	0	0	0	1
	0	0	0	1	1	0	0	0	0	1
	0	0	1	0	0	0	1	0	0	1
	0	0	1	0	1	0	1	0	0	1
	0	0	1	1	0	0	1	0	0	1
	0	0	1	1	1	0	1	0	0	1
	0	1	0	0	0	1	0	0	1	0
	0	1	0	0	1	1	0	0	1	0
	0	1	0	1	0	0	0	0	1	0
	0	1	0	1	1	0	0	0	1	0
	0	1	1	0	0	1	1	0	1	0
	0	1	1	0	1	0	1	0	1	0
	0	1	1	1	0	0	0	0	1	0
	0	1	1	1	1	0	0	0	1	0
Count	1	0	0	0	0	1	0	0	0	0
	1	0	0	0	1	1	0	0	0	0
	1	0	0	1	0	0	0	0	0	0
	1	0	0	1	1	0	0	0	0	0
	1	0	1	0	0	0	1	0	0	0
	1	0	1	0	1	0	1	0	0	0
	1	0	1	1	0	0	0	0	0	0
	1	0	1	1	1	0	0	0	0	0
Idle	1	1	0	0	0	1	0	1	0	1
	1	1	0	0	1	1	0	1	0	1
	1	1	0	1	0	0	0	1	0	1
	1	1	0	1	1	0	0	1	0	1
	1	1	1	0	0	0	1	1	0	1
	1	1	1	0	1	0	1	1	0	1
	1	1	1	1	0	0	0	1	0	1
	1	1	1	1	1	0	0	0	1	0
TC	1	1	1	1	1	1	0	0	1	0

$$n1 = (s1 + s0)cnt'clr' + s1's0*cnt*clr'sum_{lt_15}'$$

Exercícios

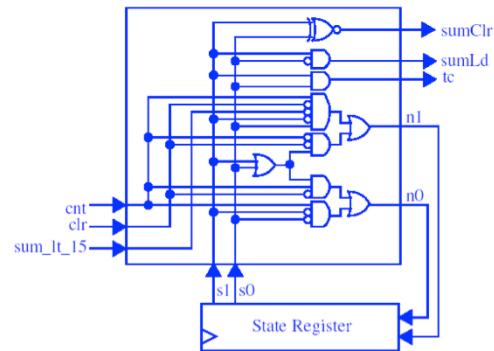
$$n0 = s1's0'cnt + (s1 + s0)cnt*clr'$$

- Resposta do 5.6
(continuação):

$$tc = s1s0$$

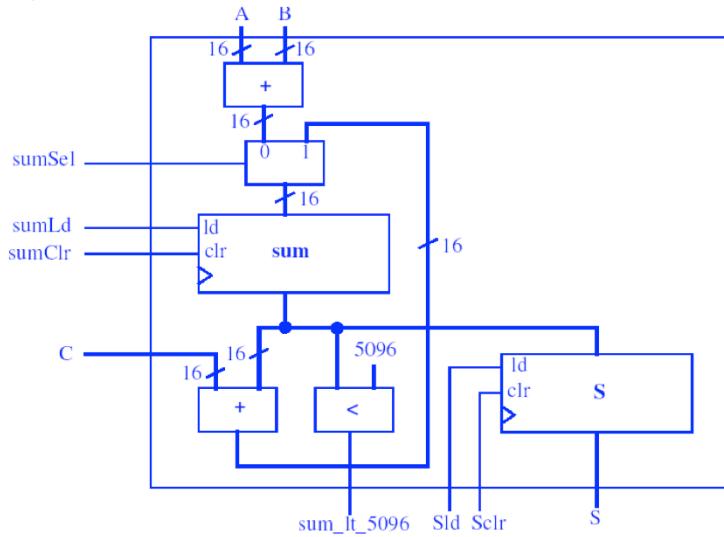
$$sumLd = s1's0$$

$$sumClr = s1's0' + s1s0$$



Exercícios

- Resposta do 5.8:



Exercícios

- Resposta do 5.10:

Inputs: start, w_wait (bit)
Outputs: w_wr, w_addr_ld, w_data_ld (bit)

