



Universidade Federal de Santa Catarina

Centro Tecnológico

Departamento de Informática e Estatística
Ciências da Computação e Engenharia Eletrônica



Sistemas Digitais

INE 5406

Aula 5-T

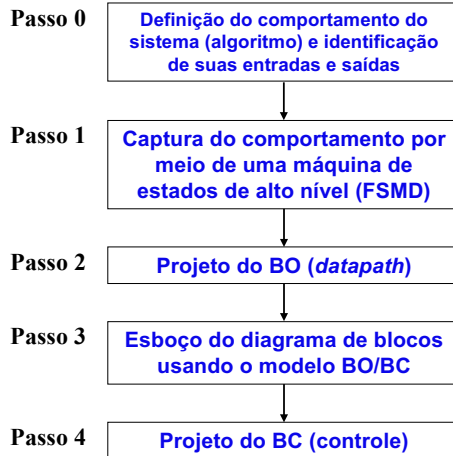
**2. Processadores Dedicados (Blocos Aceleradores). Método de Projeto no Nível RT. Exemplos 5, 6 e 7 (multiplicação sequencial).
Exploração do Espaço de Soluções: comparações de desempenho e custo.**

Profs. José Luís Güntzel e Cristina Meinhardt

{j.guntzel, cristina.meinhardt}@ufsc.br

Processadores Dedicados

Método de Projeto de Sistemas Digitais no Nível RT



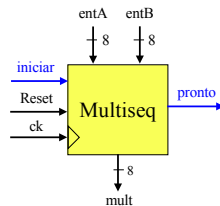
Processadores Dedicados

Exemplo 5: Enunciado

SD para multiplicação sequencial de dois inteiros sem sinal

Especificação:

- O sistema digital “Multseq” deve calcular, de maneira sequencial, a multiplicação entre dois números inteiros sem sinal **A** e **B**, representados com 8 bits;
- Há dois sinais de controle externos: “iniciar” e “pronto”.
- Há duas entradas de dados de 8 bits, “entA” e “entB”, pelas quais **A** e **B** são fornecidos simultaneamente, no momento adequado;
- O resultado é fornecido pela saída “mult”, representado com 8 bits, e deve ficar disponível mesmo enquanto um novo cálculo é realizado. Suponha que **A** e **B** sejam tais que o resultado de $A \times B < 256$. Adote um algoritmo apropriado...



Novidade!

Requisitos:

- O **custo** de implementação do B.O. deve ser mínimo;
- Não há restrição quanto ao **desempenho** mínimo necessário.

Processadores Dedicados

Exemplo 5: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial de dois inteiros sem sinal


Escolha do Algoritmo*:

Solução mais ingênua → escolher um operando para servir de multiplicador e realizar somas sucessivas usando o outro operando (enquanto o multiplicador vai sendo decrementado até atingir zero).

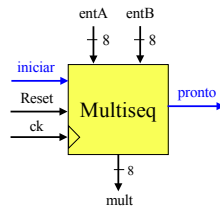
Exemplo:

$$\begin{array}{c} \text{multiplicador} \nearrow \\ A \times B = 3 \times 4 = 4 + 4 + 4 = 12 \\ \nwarrow \text{multiplicando} \end{array}$$

somas sucessivas



Problema: o número de operações de soma é dependente do valor dos operandos, especialmente do valor do multiplicador

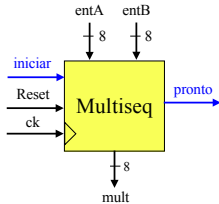


```
0. Início
1. A ← entA; B ← entB; P ← 0; pronto ← 0;
2. Se B ≠ 0 então
3. Enquanto A ≠ 0 faça {
4.   P ← P + B;
5.   A ← A - 1; }
6. mult ← P;
7. pronto ← 1;
```

* Faz parte do passo 0 do método de projeto no nível RT

Processadores Dedicados

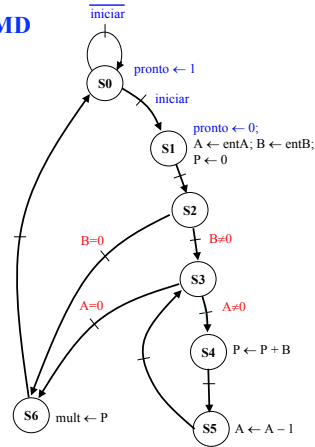
Exemplo 5: Passo 1 (Captura do comportamento por meio de uma FSM)



0. Início
1. $A \leftarrow entA$; $B \leftarrow entB$; $P \leftarrow 0$; $pronto \leftarrow 0$;
2. **Se** $B \neq 0$ **então**
3. **Enquanto** $A \neq 0$ **faça** {
4. $P \leftarrow P + B$;
5. $A \leftarrow A - 1$; }
6. $mult \leftarrow P$;
7. $pronto \leftarrow 1$;

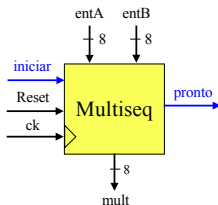


FSMD



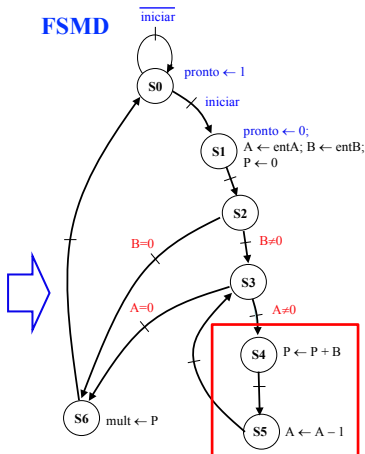
Processadores Dedicados

Exemplo 5: Passo 1 (Captura do comportamento por meio de uma FSM)



0. Início
 1. $A \leftarrow \text{entA}$; $B \leftarrow \text{entB}$; $P \leftarrow 0$; **pronto $\leftarrow 0$** ;
 2. **Se $B \neq 0$ então**
 3. **Enquanto $A \neq 0$ faça {**
 4. $P \leftarrow P + B$;
 5. $A \leftarrow A - 1$; }
 6. $\text{mult} \leftarrow P$;
 7. **pronto $\leftarrow 1$** ;

FSMD



Componente RT	Custo
Somador	24n
Subtrator	26n
Somador/subtrator	30n

Alocar estas duas operações em estados distintos permite que se use um único somador/subtrator (custo = 30 transistores por bit), ao invés de se usar um somador e um subtrator (custo = 24+26=50 transistores por bit), o que auxilia a minimizar o custo do BO, conforme especificado no enunciado deste exemplo.

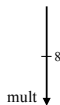
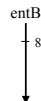
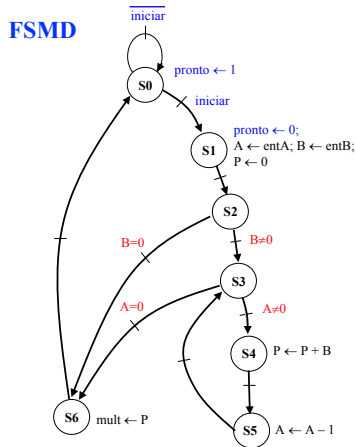
Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO)

1ª questão para guiar o projeto do BO:

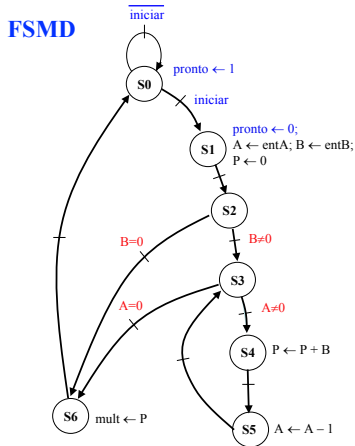
Quais são os sinais de interface do BO?

- “entA”, “entB” e “mult”



Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO)



2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” e “mult”
- Quantos registradores para armazenar tais variáveis?

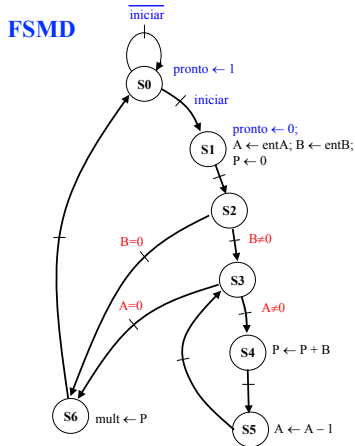
Análise do tempo de vida das variáveis:

	0	1	2	3	4	5	6
A			X	X	X	X	
B			X	X	X	X	
P			X	X	X	X	X
mult	X	X	X	X	X	X	X

Uma variável está “viva” desde o ciclo de relógio subsequente àquele em que ela recebe um valor novo até o último ciclo no qual ela é consultada.

Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO)



2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” e “mult”
- Quantos registradores para armazenar tais variáveis?

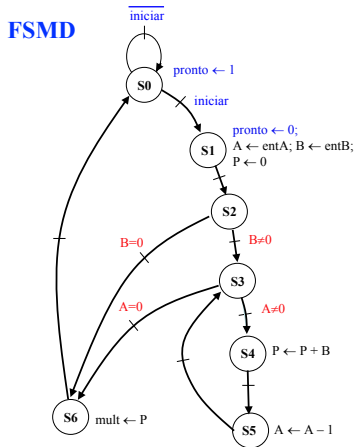
Análise do tempo de vida das variáveis:

	0	1	2	3	4	5	6
A			X	X	X	X	
B			X	X	X	X	
P			X	X	X	X	X
mult	X	X	X	X	X	X	X

“mult” está sempre viva porque seu objetivo é armazenar o último resultado, mesmo enquanto “multiseq” executa um novo cálculo.

Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO)



2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” e “mult”
- Quantos registradores para armazenar tais variáveis?

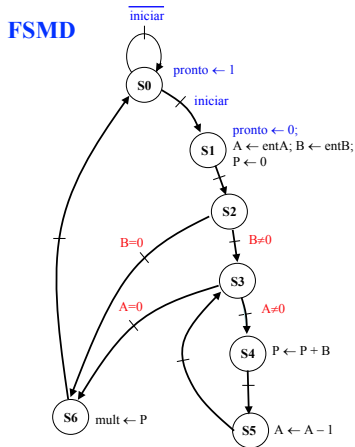
Análise do tempo de vida das variáveis:

	0	1	2	3	4	5	6
A			X	X	X	X	
B			X	X	X	X	
P			X	X	X	X	X
mult	X	X	X	X	X	X	X

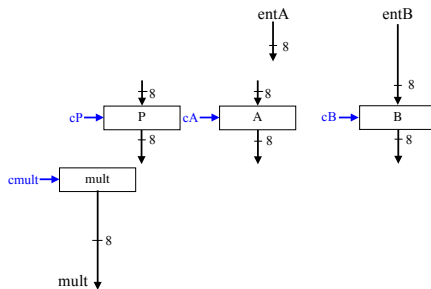
O número máximo de variáveis simultaneamente “vivas” é 4.

Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO)



Logo, são necessários **4 registradores**, exclusivos para armazenar cada variável: A, B, P e mult.

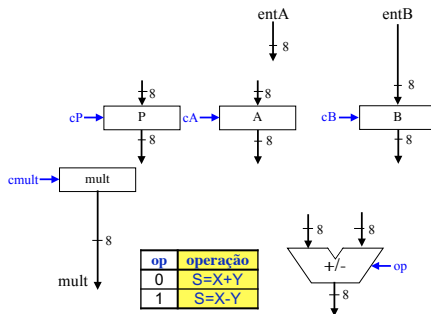
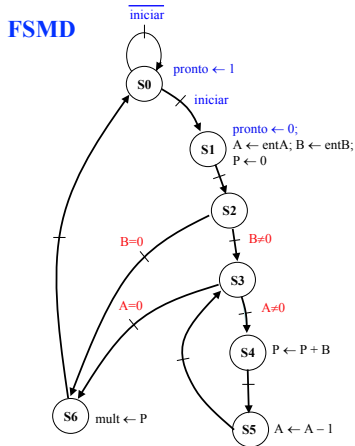


Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO) 3ª questão para guiar o projeto do BO:

Quais operações são realizadas?

- Uma adição e uma subtração
- Como a adição e a subtração são realizadas em estados distintos, usaremos um somador/subtrator para minimizar o custo do B.O.



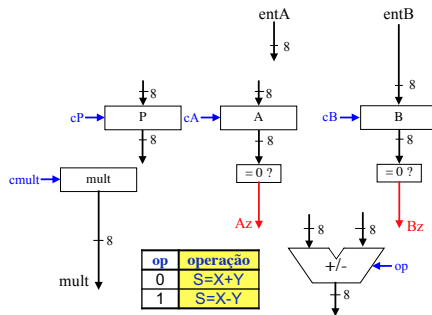
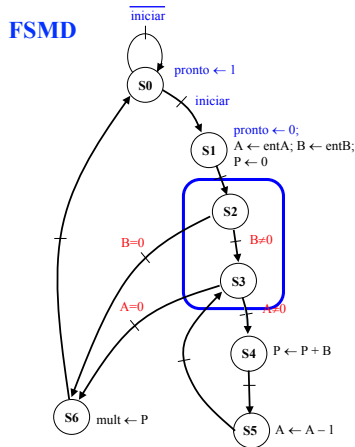
Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO) 3ª questão para guiar o projeto do BO:

Quais operações são realizadas?

- Também há 2 testes: “A=0?” e “B=0?”
- Usaremos dois testadores, pois seu custo é baixo

→ Logo, poderemos realizar os testes **A≠0** e **B≠0** no mesmo estado



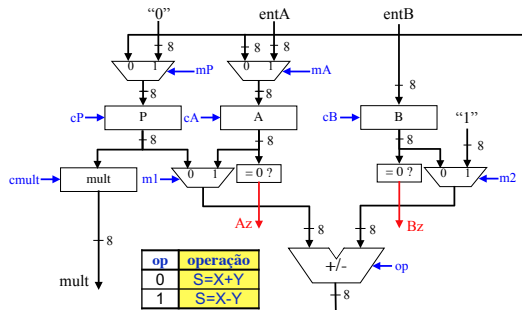
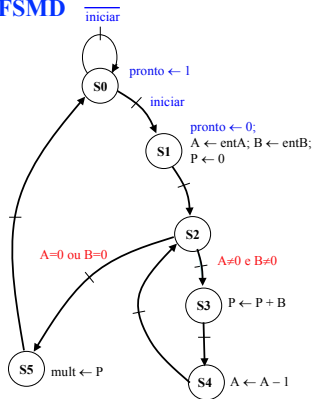
Processadores Dedicados

Exemplo 5: Passo 2 (Projeto do BO) 4ª questão para guiar o projeto do BO:

Quais conexões? Variáveis x operações

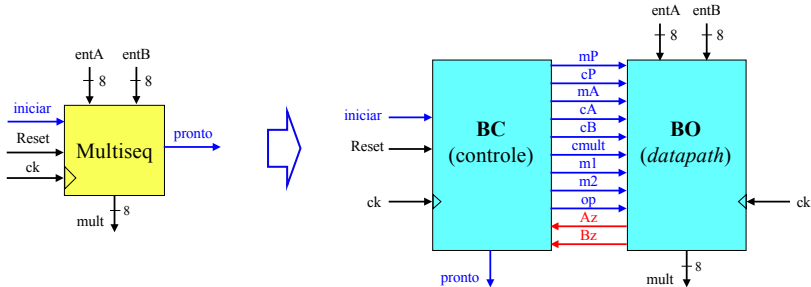
- $P \leftarrow 0$, $P \leftarrow P+B$; $A \leftarrow \text{entA}$; $A \leftarrow A-1$; $B \leftarrow \text{entB}$; $\text{mult} \leftarrow P$;
- Deve haver mux2:1 nas entradas de P, A e +/-

Nova FSMD



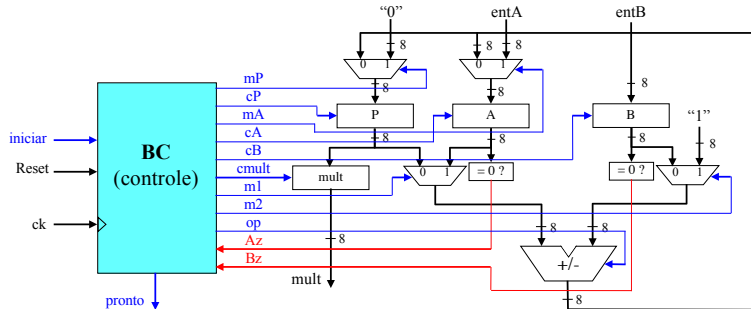
Processadores Dedicados

Exemplo 5: Passo 3 (Esboçando o Diagrama BO/BC)



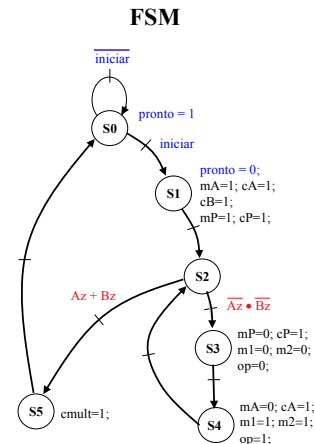
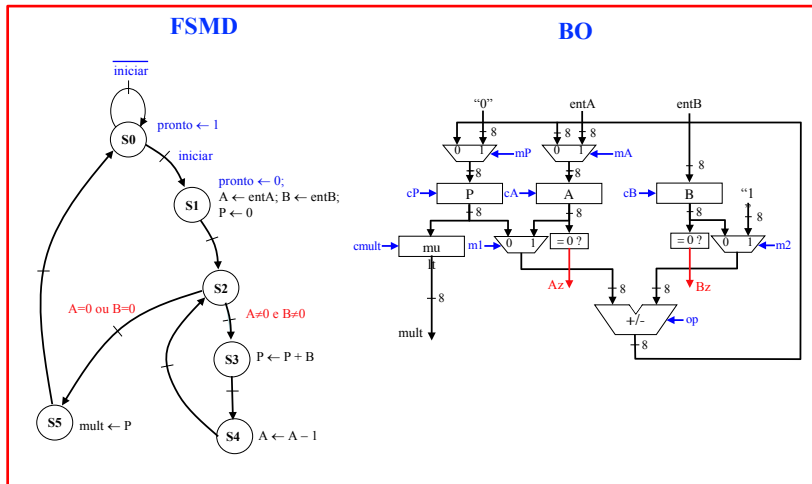
Processadores Dedicados

Exemplo 5: Passo 3 (Esboçando o Diagrama BO/BC)



Processadores Dedicados

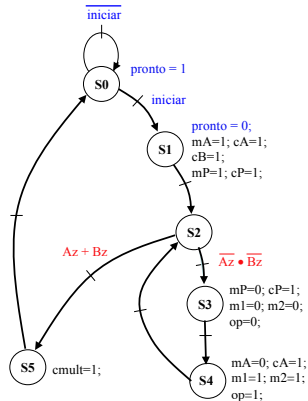
Exemplo 5: Passo 4 (Projeto do BC): Criando uma FSM a partir da FSMD e do BO



Processadores Dedicados

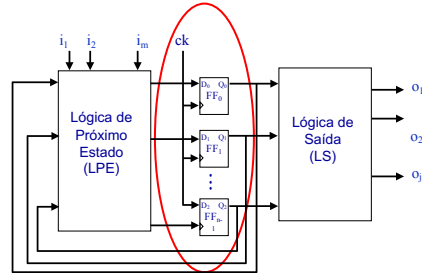
Exemplo 5: Passo 4 (Projeto do BC): Definindo o número de flip-flops

FSM



Quantos flip-flops são necessários para implementar a FSM?

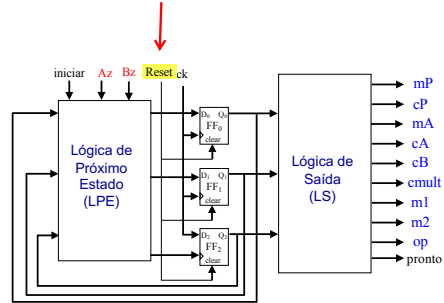
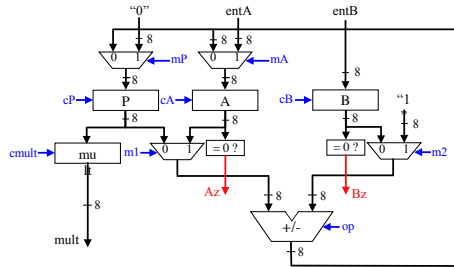
Resp.: como são **6** estados (=6 combinações), o cálculo do número de flip-flops é $\log_2 6 = 3$



Processadores Dedicados

Exemplo 5: Passo 4 (Projeto do BC): Mapeando as interfaces do BC para o modelo de FSM de Moore

Sugestão: tal como nos Exemplos anteriores, codificar o estado S0 como "000", uma vez que ele será o estado de Reset.



Processadores Dedicados

Exemplo 5: Passo 4 (Projeto do BC): projetando a Lógica de Próximo Estado (LPE)

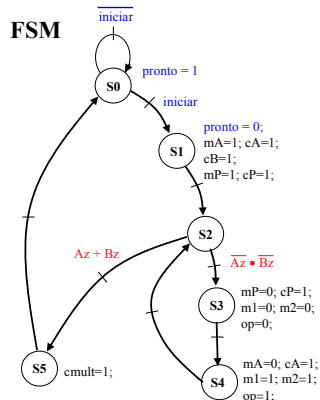


Tabela de Transição de Estados (Assumindo Moore)

Estado atual	Entradas			Próx. Estado
	iniciar	Bz	Az	
S0	0	-	-	S0
	1	-	-	S1
S1	-	-	-	S2
	-	0	0	S3
	-	0	1	S5
	-	1	0	S5
S2	-	1	1	S5
	-	-	-	S4
S3	-	-	-	S4
S4	-	-	-	S2
S5	-	-	-	S0

Processadores Dedicados

Exemplo 5: Passo 4 (Projeto do BC): projetando a Lógica de Saída (LS)

FSM

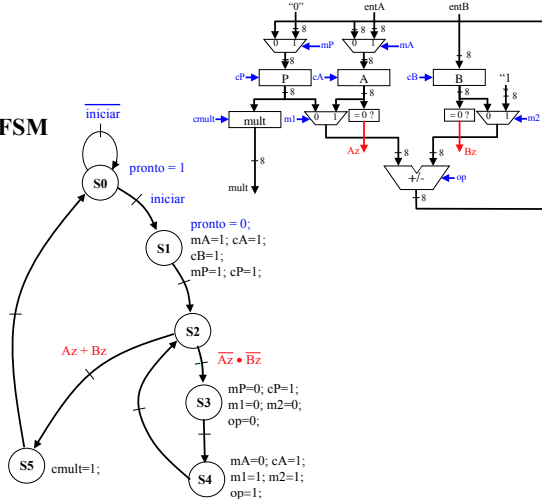


Tabela de Saídas (Assumindo modelo de Moore)

Estado	Reg. P		Reg. A			Somador/Sub			Saída	
	mP	cP	mA	cA		m1	m2	op	cmult	pronto
S0	-	0	-	0	0	-	-	-	0	1
S1	1	1	1	1	1	-	-	-	0	0
S2	-	0	-	0	0	-	-	-	0	0
S3	0	1	-	0	0	0	0	0	0	0
S4	-	0	0	1	0	1	1	1	0	0
S5	-	0	-	0	0	-	-	-	1	0

1 sinal

1 sinal

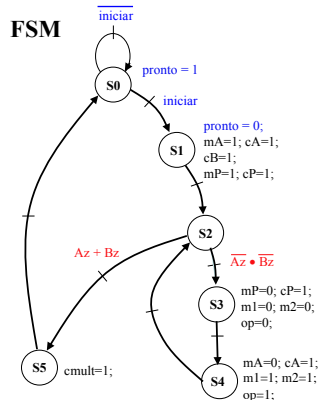
$mP = mA = CB$
 CP
 $CA = op = m1 = m2$
 $cmult$
 $pronto$

5 sinais

OBS:
na tabela, don't cares estão representados por "-"

Processadores Dedicados

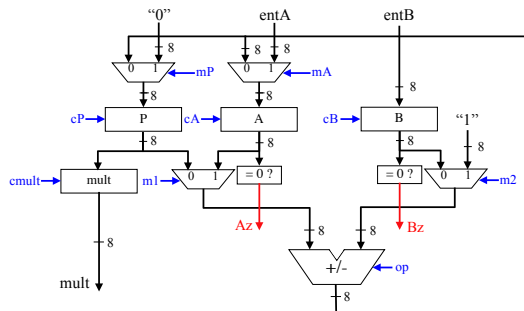
Exemplo 5: Passo 4 (Projeto do BC)



Os passos faltantes são similares aos Exemplos 1, 2, 3 e 4

Processadores Dedicados

Exemplo 5: Estimativa de Custo para o BO (e para o BC)



Componente do B.O.	Custo
1 Somador/subtrator de n bits	30n
4 Muxes 2:1 de n bits	4x4n=16n
4 Registradores de n bits com carga paralela controlada	4x22n=88n
2 testadores de zero ("= 0 ?")	Desprezível
Total	134n

Para $n=8 \rightarrow \text{custo} = 134 \times 8 = 1072$ transistores

Estimativa de custo para o BC:

- Número de estados da FSMD/FSM: 6
- Número de saídas distintas da LS* = 5

* = sinais de comando + sinais de controle de saída, diferentes entre si

Processadores Dedicados

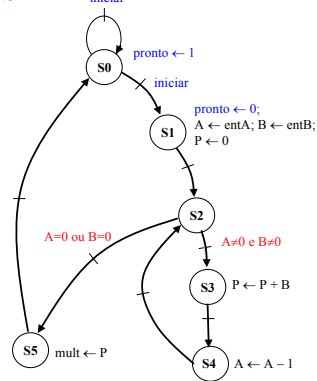
Exemplo 5: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

FSMD



Processadores Dedicados

Exemplo 5: Estimativa de Desempenho

Tempo de Execução:

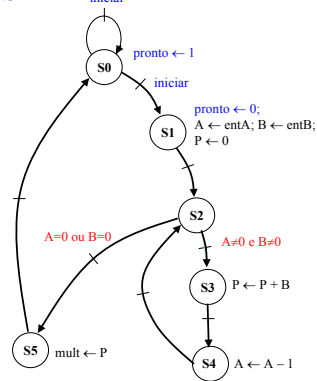
$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
- Pior caso: $A=255, B \neq 0$
- Sequência de execução: $S1, 255 \times [S2, S3, S4], S2, S5$
 \rightarrow **768 ciclos de relógio**

FSMD



Processadores Dedicados

Exemplo 5: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

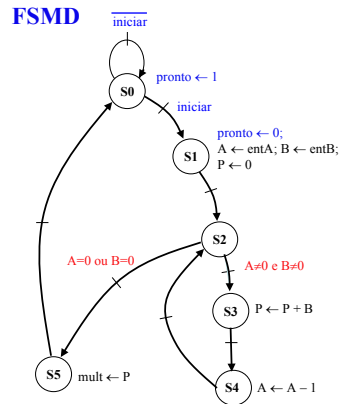
- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
- Pior caso: $A=255, B \neq 0$
- Sequência de execução: $S1, 255 \times [S2, S3, S4], S2, S5$
→ **768 ciclos de relógio**

Generalizando para n bits:

- Maior inteiro sem sinal: $2^n - 1$
- Pior caso: $A = 2^n - 1, B \neq 0$
- Sequência de execução: $S1, (2^n - 1) \times [S2, S3, S4], S2, S5$
→ $3 \times (2^n - 1) + 3$ ciclos de relógio = **3×2^n ciclos de relógio**



Falta estimar T...

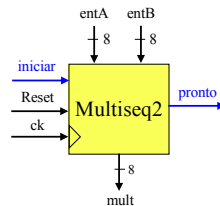
Processadores Dedicados

Exemplo 6: Enunciado

SD para multiplicação sequencial de dois inteiros sem sinal

Especificação:

- O sistema digital “**Multseq2**” deve calcular, de maneira sequencial, a multiplicação entre dois números inteiros sem sinal **A** e **B**, representados com 8 bits;
- Há dois sinais de controle externos: “iniciar” e “pronto”.
- Há duas entradas de dados de 8 bits, “entA” e “entB”, pelas quais **A** e **B** são fornecidos simultaneamente, no momento adequado;
- O resultado é fornecido pela saída “mult”, representado com 8 bits, e deve ficar disponível mesmo enquanto um novo cálculo é realizado. Suponha que **A** e **B** sejam tais que o resultado de $A \times B < 256$. Adote um algoritmo apropriado...

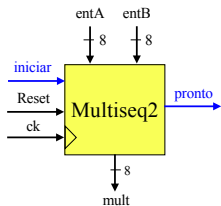


Requisitos:

- O **desempenho** deve ser máximo;
- Não há restrição quanto ao **custo** de implementação.

Processadores Dedicados

Exemplo 6: Passo 1 (Captura do comportamento por meio de uma FSMD)



0. Início

1. $A \leftarrow entA$; $B \leftarrow entB$; $P \leftarrow 0$; $pronto \leftarrow 0$;

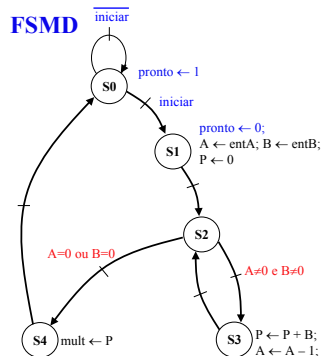
2. Se $B \neq 0$ então

3. Enquanto $A \neq 0$ faça {

4. $P \leftarrow P + B$; $A \leftarrow A - 1$; }

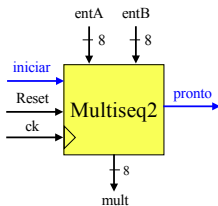
5. $mult \leftarrow P$;

6. $pronto \leftarrow 1$;

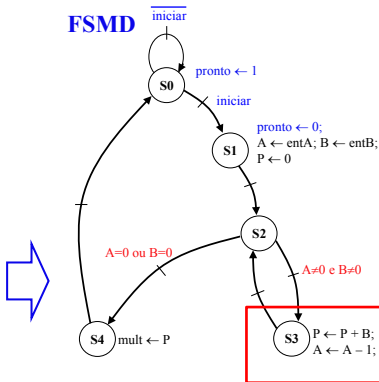


Processadores Dedicados

Exemplo 6: Passo 1 (Captura do comportamento por meio de uma FSM)



0. Início
1. $A \leftarrow \text{entA}$; $B \leftarrow \text{entB}$; $P \leftarrow 0$; **pronto $\leftarrow 0$** ;
2. **Se $B \neq 0$ então**
3. **Enquanto $A \neq 0$ faça {**
4. $P \leftarrow P + B$; $A \leftarrow A - 1$; }
5. $\text{mult} \leftarrow P$;
6. **pronto $\leftarrow 1$** ;



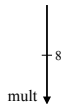
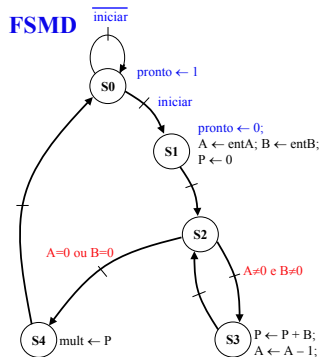
Estas duas operações serão executadas no mesmo estado (portanto, em paralelo), a fim de aumentar o desempenho mediante a redução do número de ciclos de relógio necessários para concluir a execução. Isso exigirá o uso de um somador e de um subtrator (portanto, aumentado o custo do BO)

Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO)

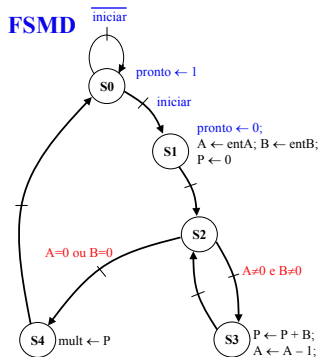
1ª questão para guiar o projeto do BO:
Quais são os sinais de interface do BO?

- “entA”, “entB” e “mult”



Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO)



2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” e “mult”
- Quantos registradores para armazenar tais variáveis?

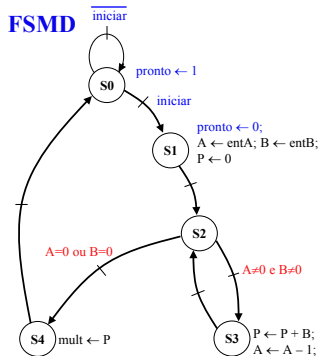
Análise do tempo de vida das variáveis:

	0	1	2	3	4
A			X	X	
B			X	X	
P			X	X	X
mult	X	X	X	X	X

“mult” está sempre viva porque seu objetivo é armazenar o último resultado, mesmo enquanto “multiseq” executa um novo cálculo.

Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO)



2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” e “mult”
- Quantos registradores para armazenar tais variáveis?

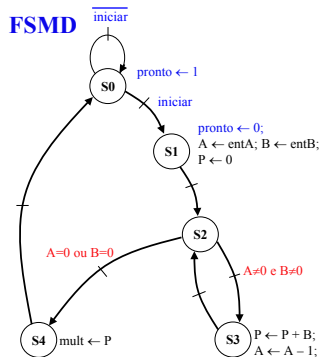
Análise do tempo de vida das variáveis:

	0	1	2	3	4
A			X	X	
B			X	X	
P			X	X	X
mult	X	X	X	X	X

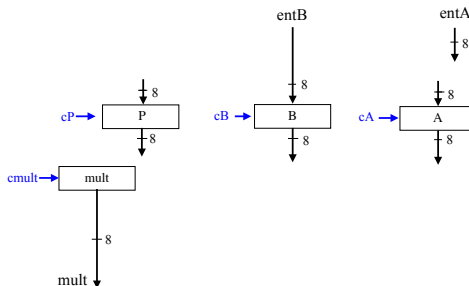
O número máximo de variáveis simultaneamente “vivas” é 4.

Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO)



Logo, são necessários **4 registradores**, exclusivos para armazenar cada variável: A, B, P e mult.

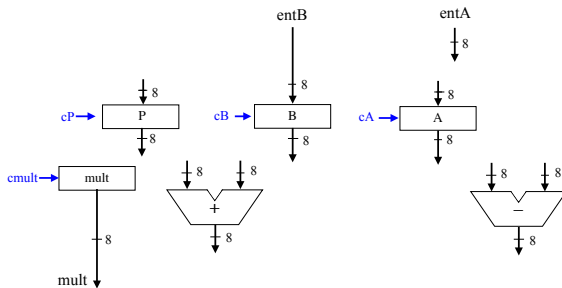
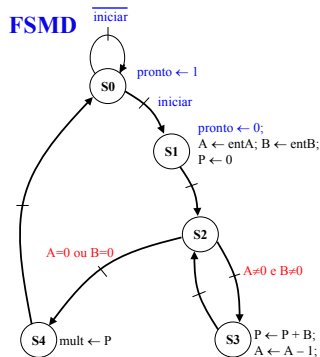


Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO) 3ª questão para guiar o projeto do BO:

Quais operações são realizadas?

- Uma adição e uma subtração, realizadas simultaneamente

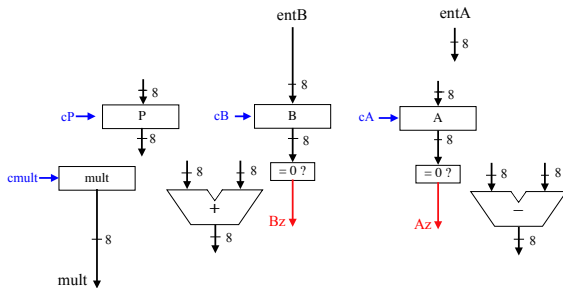
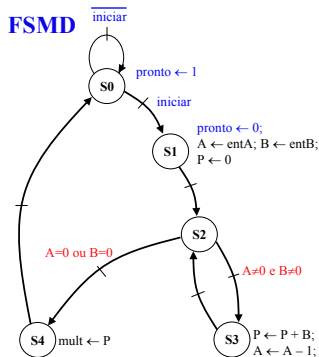


Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO) 3ª questão para guiar o projeto do BO:

Quais operações são realizadas?

- Uma adição e uma subtração, realizadas simultaneamente
- Também há 2 testes: “A=0” e “B=0”
- Usaremos dois testadores, pois seu custo é baixo

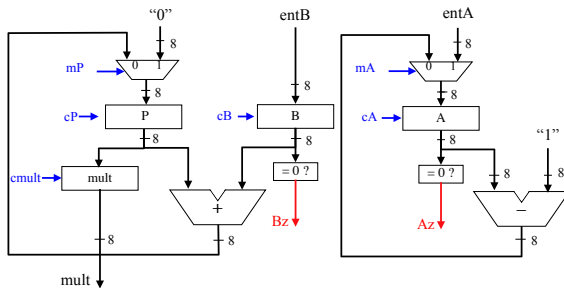
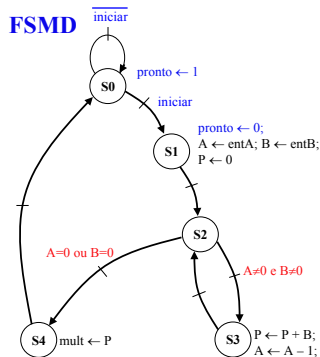


Processadores Dedicados

Exemplo 6: Passo 2 (Projeto do BO) 4ª questão para guiar o projeto do BO:

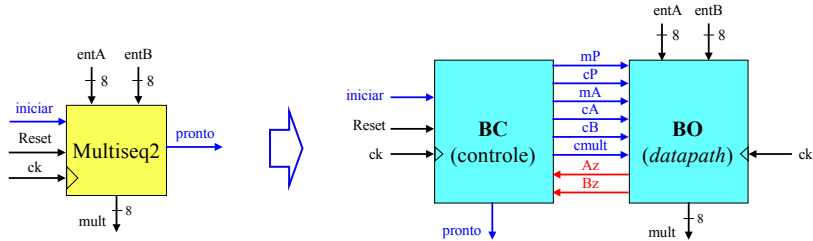
Quais conexões? Variáveis x operações

- $P \leftarrow 0$, $P \leftarrow P+B$; $A \leftarrow \text{entA}$; $A \leftarrow A-1$; $B \leftarrow \text{entB}$; $\text{mult} \leftarrow P$;
- Deve haver um mux2:1 nas entradas de P e de A



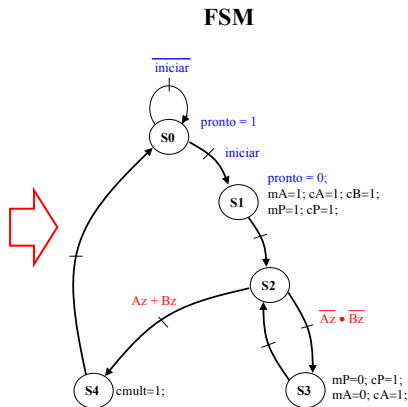
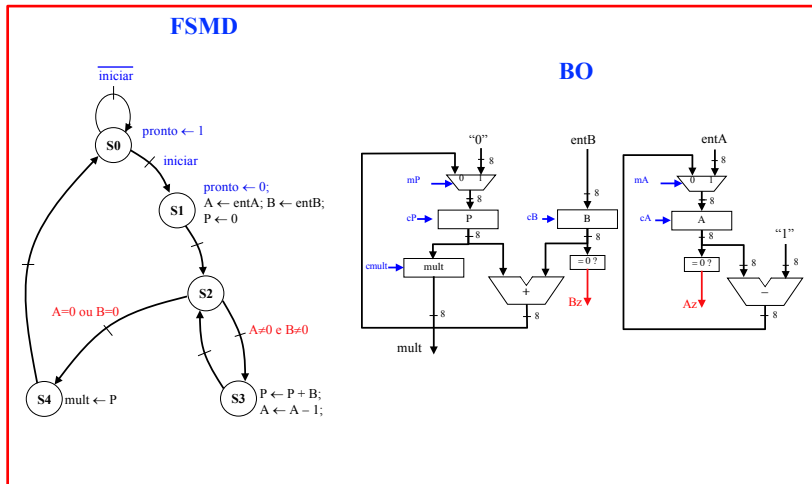
Processadores Dedicados

Exemplo 6: Passo 3 (Esboçando o Diagrama BO/BC)



Processadores Dedicados

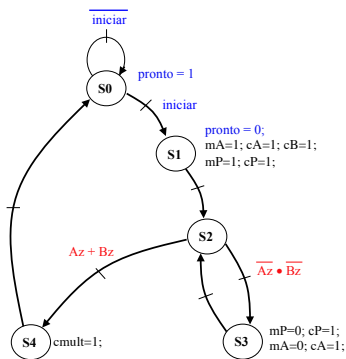
Exemplo 6: Passo 4 (Projeto do BC): Criando uma FSM a partir da FSMD e do BO



Processadores Dedicados

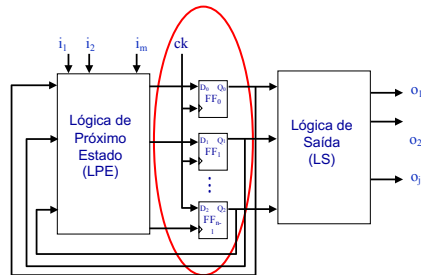
Exemplo 6: Passo 4 (Projeto do BC): Definindo o número de flip-flops

FSM



Quantos flip-flops são necessários para implementar a FSM?

Resp.: como são **5** estados (=5 combinações), o cálculo do número de flip-flops é $\log_2 5 = 3$



Processadores Dedicados

Exemplo 6: Passo 4 (Projeto do BC): projetando a Lógica de Próximo Estado (LPE)

FSM

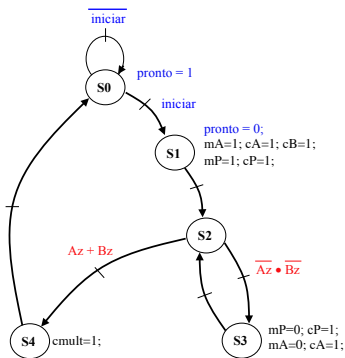


Tabela de Transição de Estados (Assumindo Moore)

Estado atual	Entradas			Próx. Estado
	iniciar	Bz	Az	
S0	0	-	-	S0
	1	-	-	S1
S1	-	-	-	S2
	-	0	0	S3
	-	0	1	S4
	-	1	0	S4
S2	-	1	1	S4
	-	-	-	S2
S3	-	-	-	S2
S4	-	-	-	S0

Processadores Dedicados

Exemplo 6: Passo 4 (Projeto do BC): projetando a Lógica de Saída (LS)

FSM

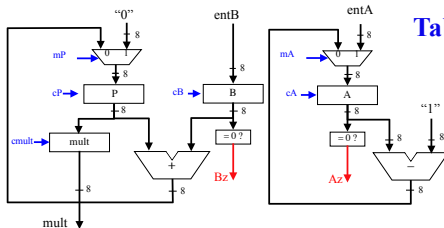
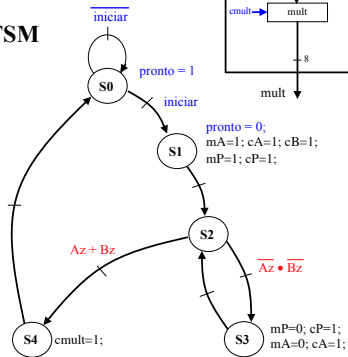


Tabela de Saídas (Assumindo modelo de Moore)

Estado	Reg. P		Reg. A			Saída	
	mP	cP	mA	cA	cB	cmult	pronto
S0	-	0	-	0	0	0	1
S1	1	1	1	1	1	0	0
S2	-	0	-	0	0	0	0
S3	0	1	0	1	0	0	0
S4	-	0	-	0	0	1	0

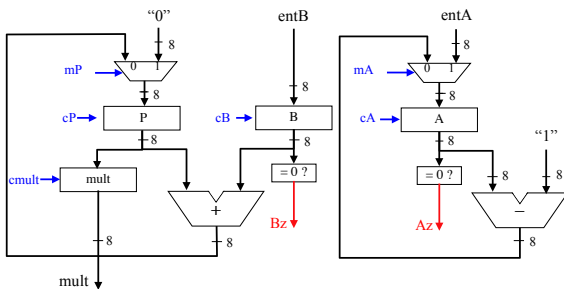
OBS: na tabela, don't cares estão representados por "-."

$\text{mP} = \text{mA} = \text{CB}$
 $\text{CP} = \text{CA}$
 cmult
 pronto

4 sinais

Processadores Dedicados

Exemplo 6: Estimativa de Custo para o BO (e para o BC)



Componente do B.O.	Custo
1 Somador de n bits	24n
1 Subtrator de n bits	26n
2 Muxes 2:1 de n bits	2x4n=8n
4 Registradores de n bits com carga paralela controlada	4x22n=88n
2 testadores de zero ("= 0 ?")	Desprezível
Total	146n

Para $n=8 \rightarrow \text{custo} = 146 \times 8 =$
1168 transistores (eram 1072)

Estimativa de custo para o BC:

- Número de estados da FSMD/FSM: **5** (eram 6)
- Número de saídas distintas da LS* = **4** (eram 5)

* = sinais de comando + sinais de controle de saída, diferentes entre si

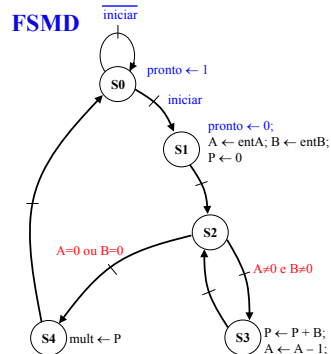
Processadores Dedicados

Exemplo 6: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio



Processadores Dedicados

Exemplo 6: Estimativa de Desempenho

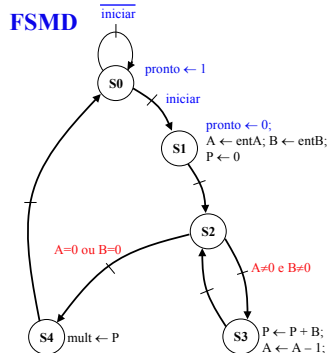
Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
 - Pior caso: $A=255, B \neq 0$
 - Sequência de execução: $S1, 255 \times [S2, S3], S2, S4$
- **513 ciclos de relógio** (BO de Mulseq: 768 ciclos)



Processadores Dedicados

Exemplo 6: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

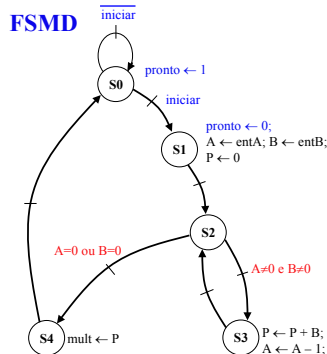
- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
- Pior caso: $A=255, B \neq 0$
- Sequência de execução: $S1, 255 \times [S2, S3], S2, S4$
→ **513 ciclos de relógio** (BO de Mulseq: 768 ciclos)

Generalizando para n bits:

- Maior inteiro sem sinal: $2^n - 1$
- Pior caso: $A = 2^n - 1, B \neq 0$
- Sequência de execução: $S1, (2^n - 1) \times [S2, S3], S2, S4$
→ $2 \times (2^n - 1) + 3 = 2 \times 2^n + 1$ **ciclos de relógio** (BO de Mulseq: 3×2^n ciclos de relógio)



Falta estimar T...

Processadores Dedicados

Comparação de Custos: Mulseq x Mulseq2

Nome do S.D.	Mulseq	Mulseq2
Característica Principal	Custo mínimo	Desempenho Máximo
Custo do BO (nº de transistores) n=8	134n 1072	146n 1168
Custo do BC nº de estados nº de sinais de controle	6 5	5 4
Nº de ciclos de relógio n=8	$\sim 3 \times 2^n$ 768	$2 \times 2^{n+1}$ 513

A exploração do paralelismo inerente ao algoritmo resultou em:

- Redução do número de ciclos de relógio. No caso estudado, a redução foi de 33%.
- Maior custo do BO. No caso estudado, +9%.
- Menor número de sinais de controle (indício de redução do custo do BC)

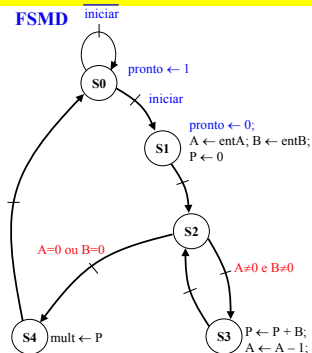
Processadores Dedicados

Exemplo 7: Enunciado

SD para multiplicação sequencial por somas e deslocamentos

O desempenho do algoritmo utilizado nos exemplos 5 e 6 depende da ordem em que os operandos são tomados...

- Considerando Mulseq2 com $n=8$ bits:
 - $A=1$ e $B=255$ (1×255) executa em **5** passos
 - $A=255$ e $B=1$ (255×1) executa em **513** passos
- Solução:** projetar outro algoritmo, tentando explorar características inerentes ao problema a ser resolvido...
- Exigência:** necessário conhecer detalhadamente o problema a ser resolvido



Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

1º Exemplo Numérico:

Com Decimais

$$\begin{array}{r} \text{x} \quad 9 \text{ multiplicando} \\ \quad 11 \text{ multiplicador} \\ \hline + \quad 9 \\ \quad 9 - \end{array} \left. \vphantom{\begin{array}{r} \text{x} \quad 9 \\ \quad 11 \\ \hline + \quad 9 \\ \quad 9 - \end{array}} \right\} \text{produtos parciais}$$

$$\hline 99 \text{ resultado}$$

Com Binários

$$\begin{array}{r} \text{x} \quad 1001 \text{ multiplicando} \\ \quad 1011 \text{ multiplicador} \\ \hline \quad 1001 \\ + \quad 1001 - \\ \quad 0000 - - \\ \quad 1001 - - - \end{array} \left. \vphantom{\begin{array}{r} \text{x} \quad 1001 \\ \quad 1011 \\ \hline \quad 1001 \\ + \quad 1001 - \\ \quad 0000 - - \\ \quad 1001 - - - \end{array}} \right\} \text{produtos parciais}$$

$$\hline 1100011 \text{ resultado}$$

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD) SD para multiplicação sequencial por somas e deslocamentos

1º Exemplo Numérico:

Seja n = nº de bits do multiplicador e m = nº de bits do multiplicando;

- Gerar n produtos parciais (de m bits)
- Calcular a soma de n produtos parciais com esquema monociclo ou com esquema multiciclo.
- Se usar multiciclo:
 - Inicializar uma variável acumuladora com zero
 - Calcular n adições de m bits, considerando o *carry out* para evitar *overflow*.

	1 0 0 1	multiplicando
x	1 0 1 1	multiplicador
<hr/>		
	1 0 0 1	} produtos parciais
+	1 0 0 1 -	
	0 0 0 0 - -	
	1 0 0 1 - - -	
<hr/>		
	1 1 0 0 0 1 1	resultado

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

	1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1	A = 5 (= Multiplicador)
	0	0	0	0	P (é um acumulador)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

	1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1	A = 5 (= Multiplicador)
<hr/>					
	0	0	0	0	P
	1	0	0	1	Prod. parcial 0 (= A(0).B<<0)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

	1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1	A = 5 (= Multiplicador)
<hr/>					
	0	0	0	0	P
+	1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
	1	0	0	1	P

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

	1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1	A = 5 (= Multiplicador)
<hr/>					
	0	0	0	0	P
+	1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
<hr/>					
	1	0	0	1	P
	0	0	0	0	Prod. Parcial 1 (= A(1).B<<1)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

		1	0	0	1	B = 9 (= Multiplicando)	
	x	0	1	0	1	A = 5 (= Multiplicador)	
<hr/>							
		0	0	0	0	P	
	+	1	0	0	1	Prod. parcial 0 (= A(0).B<<0)	
<hr/>							
		1	0	0	1	P	
	+	0	0	0	0	-	
		0	1	0	0	1	Prod. Parcial 1 (= A(1).B<<1)
		0	1	0	0	1	P

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

		1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1		A = 5 (= Multiplicador)
	0	0	0	0		P
+	1	0	0	1		Prod. parcial 0 (= A(0).B<<0)
	1	0	0	1		P
+	0	0	0	0	-	Prod. Parcial 1 (= A(1).B<<1)
	0	1	0	0	1	P
	1	0	0	1	-	Prod. Parcial 2 (= A(2).B<<2)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

		1	0	0	1	B = 9 (= Multiplicando)		
x		0	1	0	1	A = 5 (= Multiplicador)		
<hr/>								
		0	0	0	0	P		
+		1	0	0	1	Prod. parcial 0 (= A(0).B<<0)		
<hr/>								
		1	0	0	1	P		
+		0	0	0	0	-	Prod. Parcial 1 (= A(1).B<<1)	
<hr/>								
		0	1	0	0	1	P	
+	1	0	0	1	-	-	Prod. Parcial 2 (= A(2).B<<2)	
<hr/>								
		1	0	1	1	0	1	P

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

		1	0	0	1		B = 9 (= Multiplicando)
x		0	1	0	1		A = 5 (= Multiplicador)
		0	0	0	0		P
+		1	0	0	1		Prod. parcial 0 (= A(0).B<<0)
		1	0	0	1		P
+	0	0	0	0	-		Prod. Parcial 1 (= A(1).B<<1)
	0	1	0	0	1		P
+	1	0	0	1	-	-	Prod. Parcial 2 (= A(2).B<<2)
	1	0	1	1	0	1	P
	0	0	0	0	-	-	Prod. Parcial 3 (= A(3).B<<3)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

				1	0	0	1	B = 9 (= Multiplicando)		
	x			0	1	0	1	A = 5 (= Multiplicador)		
				0	0	0	0	P		
	+			1	0	0	1	Prod. parcial 0 (= A(0).B<<0)		
				1	0	0	1	P		
	+			0	0	0	0	-	Prod. Parcial 1 (= A(1).B<<1)	
				0	1	0	0	1	P	
	+			1	0	0	1	-	-	Prod. Parcial 2 (= A(2).B<<2)
				1	0	1	1	0	1	P
+		0	0	0	0	0	-	-	-	Prod. Parcial 3 (= A(3).B<<3)
		0	0	1	0	1	1	0	1	P= 45 = resultado

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

As adições podem considerar apenas o número de bits do multiplicador

					1	0	0	1	B = 9 (= Multiplicando)
	x				0	1	0	1	A = 5 (= Multiplicador)
					0	0	0	0	P
	+				1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
					1	0	0	1	P
	+				0	0	0	0	-
					0	1	0	0	1
	+				1	0	0	1	-
					1	0	1	1	0
	+				0	0	0	0	-
					0	0	1	0	1
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0
					1	0	1	1	0

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

Após cada nova
adição, um novo bit
mas à direita torna-
se definitivo

				1	0	0	1	B = 9 (= Multiplicando)
	x			0	1	0	1	A = 5 (= Multiplicador)
				0	0	0	0	P
	+			1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
				1	0	0	1	P
	+			0	0	0	0	Prod. Parcial 1 (= A(1).B<<1)
				0	1	0	0	P
	+			1	0	0	1	Prod. Parcial 2 (= A(2).B<<2)
				1	0	1	1	P
	+			0	0	0	0	Prod. Parcial 3 (= A(3).B<<3)
0	0	1	0	1	1	0	1	P = 45 = resultado

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

Porém, pode ocorrer *overflow* em qualquer adição

	1	0	0	1	B = 9 (= Multiplicando)
x	0	1	0	1	A = 5 (= Multiplicador)
	0	0	0	0	P
	1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
	1	0	0	1	P
	0	0	0	0	- Prod. Parcial 1 (= A(1).B<<1)
	0	1	0	0	0 1 P
	1	0	0	1	- - Prod. Parcial 2 (= A(2).B<<2)
	1	0	1	1	0 1 0 1 P
	0	0	0	0	- - - Prod. Parcial 3 (= A(3).B<<3)
	0	0	1	0	1 1 0 1 P = 45 = resultado

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

Porém, pode ocorrer *overflow* em qualquer adição

				1	0	0	1	B = 9 (= Multiplicando)
x				0	1	0	1	A = 5 (= Multiplicador)
				0	0	0	0	P
	+			1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
		C _{out}		1	0	0	1	P
			+	0	0	0	0	-
				0	1	0	0	1
				1	0	0	1	-
				1	0	0	1	-
				1	0	1	1	0
				0	0	0	0	-
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1
				0	1	0	1	1

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

2º Exemplo Numérico: 5 x 9 em binário, usando soma sequencial

No exemplo ao lado, $n=m$ e portanto, o resultado cabe em $2n$ (s/ overflow)
Logo, P tem $2n$ bits.

Dividiremos P em PH (n bits mais significativos) e PL (n bits menos significativos).

Algoritmo

$P \leftarrow 0$; $n \leftarrow n^\circ$ de bits dos operandos

Enquanto $n > 0$ faça

- {
- 1. $PH \leftarrow PH + \text{Prod. Parcial } n$
- 2. $P \leftarrow P \gg 1$;
- 3. $n \leftarrow n - 1$;
- }

Obs: quando (Prod. Parcial n) = 0, passo 1 não precisa ser feito.

			1	0	0	1	B = 9 (= Multiplicando)
x			0	1	0	1	A = 5 (= Multiplicador)
			0	0	0	0	P
	+		1	0	0	1	Prod. parcial 0 (= A(0).B<<0)
		C _{out}	1	0	0	1	P
			0	0	0	0	-
	+		0	1	0	0	1
		0	0	1	0	0	1
			1	0	0	1	-
	+		1	0	0	1	Prod. Parcial 2 (= A(2).B<<2)
		C _{out}	1	0	1	1	0
			0	0	0	0	-
	+		0	0	0	0	Prod. Parcial 3 (= A(3).B<<3)
		C _{out}	0	1	0	1	1
			0	1	0	1	1
			1	0	0	1	P = 45 = resultado

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD) SD para multiplicação sequencial por somas e deslocamentos

Formalizando o algoritmo:

```
início
  pronto ← 0;
  A ← entA; B ← entB; P ← 0; cont ← n; // numero de bits de A
  Se (A ≠ 0 E B ≠ 0) então
    Enquanto cont ≠ 0 faça
    {
      Se A(0)=1
        PH ← PH + B; // soma B à parte alta de P
        P ← P >> 1; // desloca P à direita de 1 bit, o novo bit à esquerda pode ser Cout ou 0
        A ← A >> 1; // desloca A à direita de 1 bit, o novo bit que entra pode ter X
        cont ← cont - 1; // incrementa contador
    }
  mult ← P;
  pronto ← 1;
fim
```


Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $\text{entA}=5$ e $\text{entB}=9$ (5×9) e portanto, ao final, $P=45$

```
início
pronto ← 0;
0 A ← entA; B ← entB; P ← 0; cont ← n;
1 Se (A ≠ 0 E B ≠ 0) então
2   Enquanto cont ≠ 0 faça
3     Se A(0) = 1
4       PH ← PH + B;
5       P ← P >> 1;
5       A ← A >> 1;
5       cont ← cont - 1;
  }
6 mult ← P;
7 pronto ← 1;
fim
```

Analisaremos estes 2 testes como se fossem executados em paralelo

Analisaremos estes 3 comandos como se fossem executados em paralelo

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)

```
início
pronto ← 0;
0 A ← entA; B ← entB; P ← 0; cont ← n;
1 Se (A ≠ 0 E B ≠ 0) então
2   Enquanto cont ≠ 0 faça
3   {
4     Se A(0) = 1
5     PH ← PH + B;
5     P ← P >> 1;
5     A ← A >> 1;
5     cont ← cont - 1;
3   }
6 mult ← P;
7 pronto ← 1;
fim
```

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que **entA=5** e **entB=9** (5×9) e portanto, ao final, $P=45$

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)

```
início
pronto ← 0;
0 A ← entA; B ← entB; P ← 0; cont ← n;
1 Se (A ≠ 0 E B ≠ 0) então
2   Enquanto cont ≠ 0 faça
3   {
4     Se A(0) = 1
5     PH ← PH + B;
6     P ← P >> 1;
7     A ← A >> 1;
8     cont ← cont - 1;
9   }
10 mult ← P;
11 pronto ← 1;
fim
```

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
      {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
      }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5      P ← P >> 1;
5      A ← A >> 1;
5      cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $\text{entA}=5$ e $\text{entB}=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
pronto ← 0;
0 A ← entA; B ← entB; P ← 0; cont ← n;
1 Se (A ≠ 0 E B ≠ 0) então
2   Enquanto cont ≠ 0 faça
3     {
4       Se A(0) = 1
5         PH ← PH + B;
5         P ← P >> 1;
5         A ← A >> 1;
5         cont ← cont - 1;
6     }
6 mult ← P;
7 pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
      {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
      }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
      {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
      }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
pronto ← 0;
0 A ← entA; B ← entB; P ← 0; cont ← n;
1 Se (A ≠ 0 E B ≠ 0) então
2   Enquanto cont ≠ 0 faça
3     {
4       Se A(0) = 1
5         PH ← PH + B;
6       P ← P >> 1;
7       A ← A >> 1;
8       cont ← cont - 1;
9     }
10 mult ← P;
11 pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)
2 e 3	XXX0	1001	0101	1010	001(1)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)
2 e 3	XXX0	1001	0101	1010	001(1)
5	XXXX	1001	0010	1101	000(0)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5        P ← P >> 1;
5        A ← A >> 1;
5        cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)
2 e 3	XXX0	1001	0101	1010	001(1)
5	XXXX	1001	0010	1101	000(0)
2 e 3	XXXX	1001	0010	1101	000(0)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```
início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5      P ← P >> 1;
5      A ← A >> 1;
5      cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)
2 e 3	XXX0	1001	0101	1010	001(1)
5	XXXX	1001	0010	1101	000(0)
2 e 3	XXXX	1001	0010	1101	000(0)
6	XXXX	1001	0010	1101	000(0)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSMD)

SD para multiplicação sequencial por somas e deslocamentos

Testando algoritmo:

Suponha que $entA=5$ e $entB=9$ (5×9) e portanto, ao final, $P=45$

```

início
  pronto ← 0;
0  A ← entA; B ← entB; P ← 0; cont ← n;
1  Se (A ≠ 0 E B ≠ 0) então
2    Enquanto cont ≠ 0 faça
    {
3      Se A(0) = 1
4        PH ← PH + B;
5      P ← P >> 1;
5      A ← A >> 1;
5      cont ← cont - 1;
    }
6  mult ← P;
7  pronto ← 1;
fim
    
```

	A	B	PH	PL	cont
0	0101	1001	0000	0000	100 (4)
1	0101	1001	0000	0000	100 (4)
2 e 3	0101	1001	0000	0000	100 (4)
4	0101	1001	1001	0000	100 (4)
5	X010	1001	0100	1000	011(3)
2 e 3	X010	1001	0100	1000	011(3)
5	XX01	1001	0010	0100	010(2)
2 e 3	XX01	1001	0010	0100	010(2)
4	XX01	1001	1011	0100	010(2)
5	XXX0	1001	0101	1010	001(1)
2 e 3	XXX0	1001	0101	1010	001(1)
5	XXXX		0010 1101 ₂ = 45 ₁₀		
2 e 3	XXXX				
6	XXXX	1001	0010	1101	000(0)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

0. Início

1. $A \leftarrow \text{entA}$; $B \leftarrow \text{entB}$; $P \leftarrow 0$; $\text{cont} \leftarrow n$; **pronto** $\leftarrow 0$

2. **Se** $(A \neq 0 \text{ E } B \neq 0)$ **então**

3. **Enquanto** $\text{cont} \neq 0$ **faça** {

4. **Se** $A(0) = 1$

5. $PH \leftarrow PH + B$;

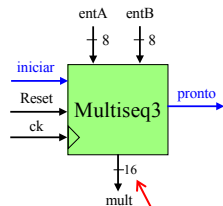
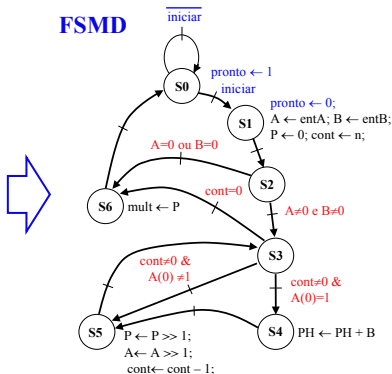
6. $P \leftarrow P \gg 1$;

7. $A \leftarrow A \gg 1$;

8. $\text{cont} \leftarrow \text{cont} - 1$; }

9. $\text{mult} \leftarrow P$;

10. **pronto** $\leftarrow 1$;



O algoritmo assume $2n$ bits para o resultado

Observações:

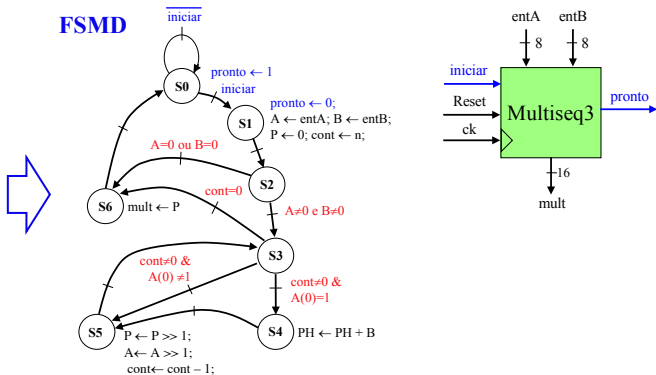
$P \leftarrow P \gg 1$; \rightarrow desloca P à direita de 1 bit, o novo bit à esquerda pode ser **Count** ou **0**

$A \leftarrow A \gg 1$; \rightarrow desloca A à direita de 1 bit, o novo bit que entra pode ser **X** (don't care)

Processadores Dedicados

Exemplo 7: Passo 1 (Captura do comportamento por meio de uma FSM)

```
0. Início
1.  $A \leftarrow \text{entA}$ ;  $B \leftarrow \text{entB}$ ;  $P \leftarrow 0$ ;  $\text{cont} \leftarrow n$ ;  $\text{pronto} \leftarrow 0$ 
2. Se  $(A \neq 0 \text{ E } B \neq 0)$  então
3. Enquanto  $\text{cont} \neq 0$  faça {
4.   Se  $A(0) = 1$ 
5.      $PH \leftarrow PH + B$ ;
6.    $P \leftarrow P \gg 1$ ;
7.    $A \leftarrow A \gg 1$ ;
8.    $\text{cont} \leftarrow \text{cont} - 1$ ; }
9.  $\text{mult} \leftarrow P$ ;
10.  $\text{pronto} \leftarrow 1$ ;
```



Se houver hardware para fazer estes dois testes simultaneamente, é possível uni-los em um único estado

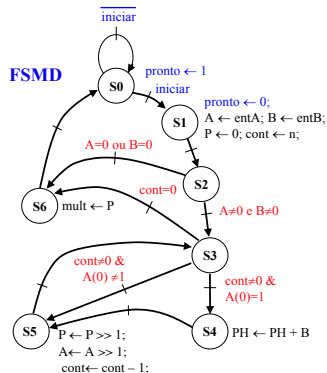
Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

1ª questão para guiar o projeto do BO:

Quais são os sinais de interface do BO?

- “entA”, “entB” e “mult”
- Usaremos 16 bits para “mult” de modo a evitar *overflow*



Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

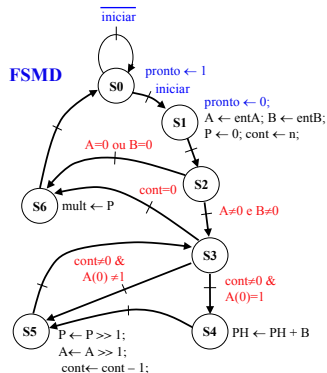
2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” (“PH” & “PL”), “cont” e “mult”
- Quantos registradores para armazenar tais variáveis?

Análise do tempo de vida das variáveis:

	0	1	2	3	4	5	6
A			X	X	X	X	
B			X	X	X	X	
P			X	X	X	X	X
cont			X	X	X	X	
mult	X	X	X	X	X	X	X



“mult” está sempre viva porque seu objetivo é armazenar o último resultado, mesmo enquanto “multiseq” executa um novo cálculo.

Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

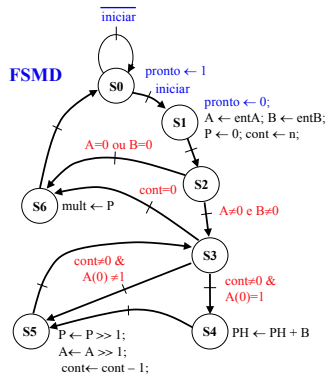
2ª questão para guiar o projeto do BO:

Quais variáveis são usadas para armazenar dados?

- “A”, “B”, “P” (“PH” & “PL”), “cont” e “mult”
- Quantos registradores para armazenar tais variáveis?

Análise do tempo de vida das variáveis:

	0	1	2	3	4	5	6
A			X	X	X	X	
B			X	X	X	X	
P			X	X	X	X	X
cont			X	X	X	X	
mult	X	X	X	X	X	X	X

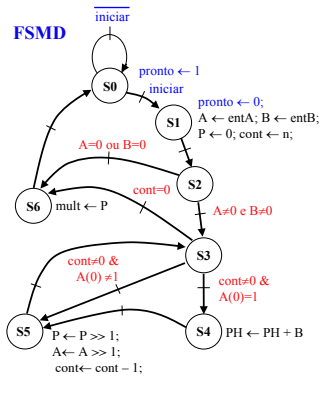
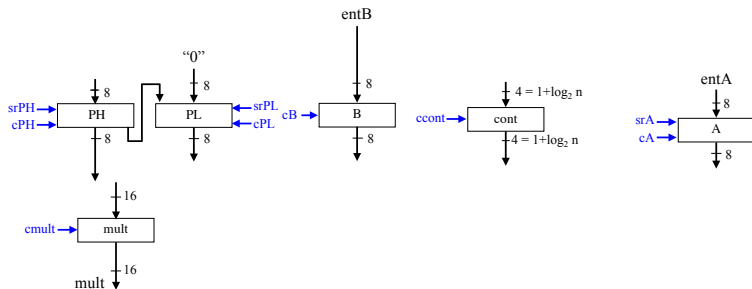


O número máximo de variáveis simultaneamente “vivas” é 5.

Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

- Logo, são necessários **5 registradores**, exclusivos para armazenar cada variável: **A, B, P, cont e mult**
- Porém, P é composto por duas parte, PH e PL, sendo possível realizar “ $P \gg 1$ ” e “ $PH + B$ ”
- PH e A podem ser deslocados à direita



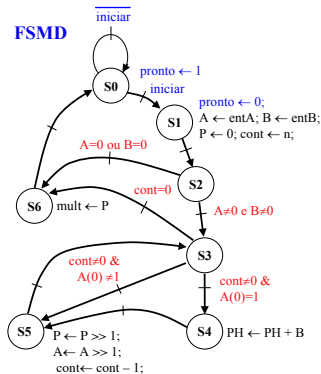
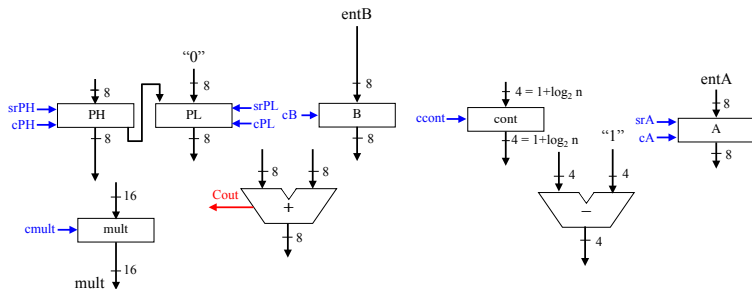
Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

3ª questão para guiar o projeto do BO:

Quais operações são realizadas

- “Uma adição, uma subtração (que poderiam usar um somador/subtrator) e deslocamentos para direita de P e A



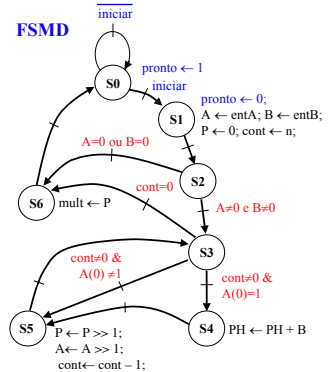
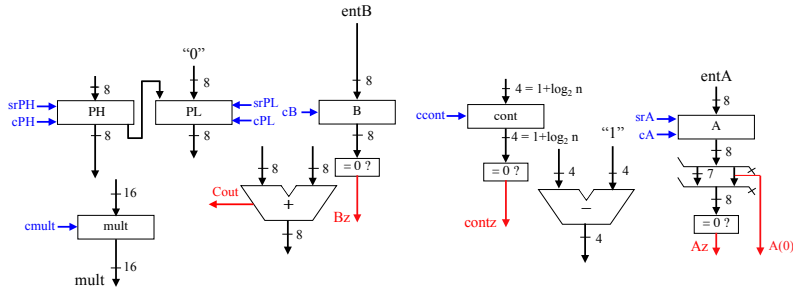
Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

3ª questão para guiar o projeto do BO:

Quais operações são realizadas

- Também há 4 testes: “A=0”, “B=0”, “cont=0” e “A(0)=0”
- Usaremos três testadores de zero e consultaremos o bit A(0)



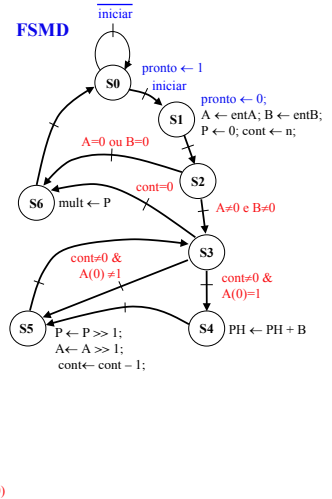
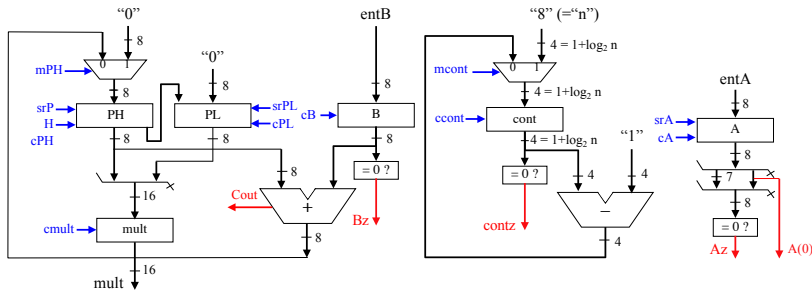
Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

4ª questão para guiar o projeto do BO:

Quais conexões? Variáveis x operações

- $P \leftarrow 0$, $PH \leftarrow PH + B$; $P \leftarrow P \gg 1$; $A \leftarrow entA$; $A \leftarrow A \gg 1$;
 $cont \leftarrow n$; $cont \leftarrow cont - 1$; $B \leftarrow entB$; $mult \leftarrow P$;
- Deve haver um mux2:1 nas entradas de PH e de cont

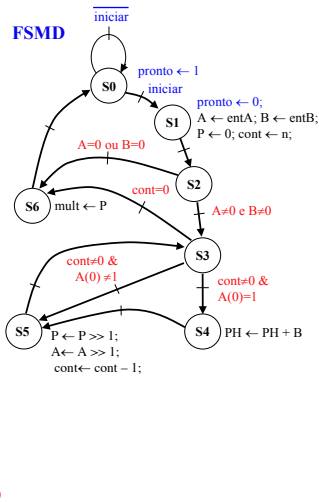
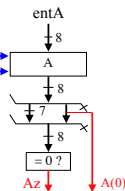
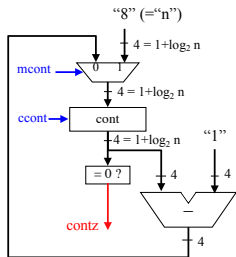
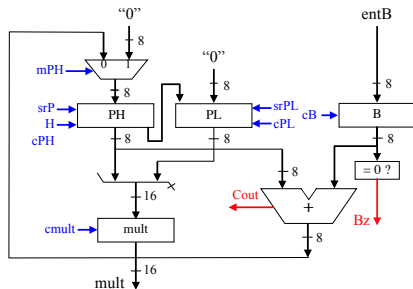


Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

Observação:

- E se ocorrer um overflow em uma operação intermediária do tipo $PH \leftarrow PH + B$?



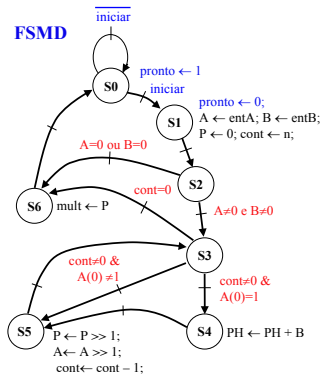
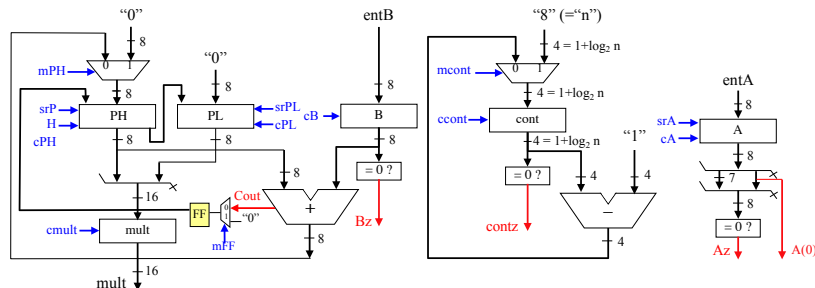
Processadores Dedicados

Exemplo 7: Passo 2 (Projeto do BO)

Observação:

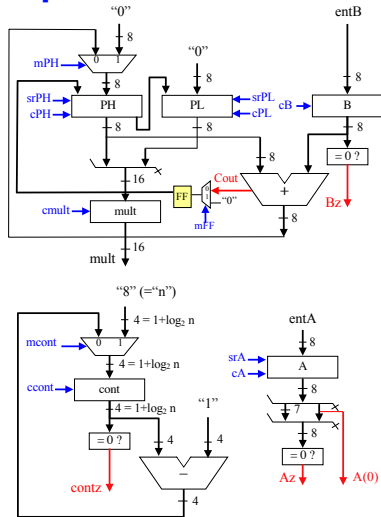
- E se ocorrer um overflow em uma operação intermediária do tipo $PH \leftarrow PH + B$?

Resposta: injetar overflow pela esquerda de “P” em S5 caso antes de S5 tenha ocorrido S4. (Senão, injetar um zero.)



Processadores Dedicados

Exemplo 7: Estimativa de Custo para o BO (e para o BC)



Componente do B.O.	Custo
1 Somador de n bits	$24n$
1 Subtrator de $1+\log_2 n$ bits	$26(1+\log_2 n)$
1 Mux 2:1 de n bits	$4n$
1 Mux 2:1 de $1+\log_2 n$ bits	$4(1+\log_2 n)$
1 Registrador de n bits com carga paralela controlada (B) e 1 Registrador de $2n$ bits com carga paralela	$3 \times 22n = 66n$
1 Registrador de $1+\log_2 n$ bits com carga paralela controlada (cont)	$22(1+\log_2 n)$
3 Registradores de n bits com carga paralela controlada e deslocamento p/ dir (A, PH e PL)	$3 \times 26n = 78n$
4 testadores de zero (" $= 0 ?$ ")	Desprezível
Total	$172n + 52(1+\log_2 n)$

Para $n=8 \rightarrow$ custo = **1584 transistores**

Estimativa de custo para o BC:

- Número de estados da FSMD/FSM: **7** (versões anteriores: 6 e 5)
- Número de saídas distintas da LS* = **4** (versões anteriores: 4 e 5)

* = sinais de comando + sinais de controle de saída, diferentes entre si

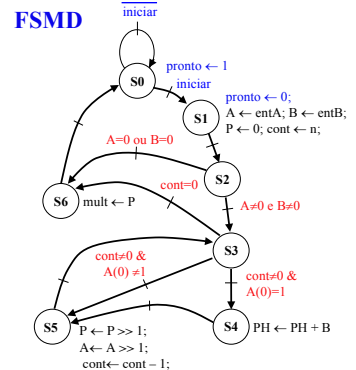
Processadores Dedicados

Exemplo 7: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio



Processadores Dedicados

Exemplo 7: Estimativa de Desempenho

Tempo de Execução:

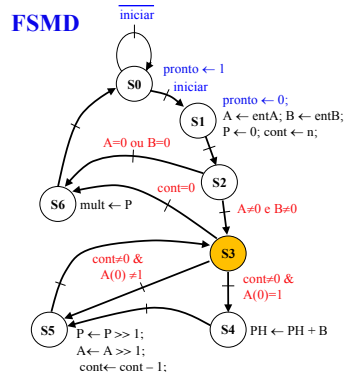
$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
- Pior caso **A=255, B $\neq 0$**
- Sequência de execução: S1, S2, 8x[S3,S4,S5], S3, S6
- **28 ciclos de relógio** (BO Mulseq: 768 ciclos, BO Mulseq2: 513 ciclos)

Para não "pular" a execução de S3 é preciso que todos os bits de A seja "1".



Processadores Dedicados

Exemplo 7: Estimativa de Desempenho

Tempo de Execução:

$$T_{\text{exec}} = n_{\text{ciclos}} \times T$$

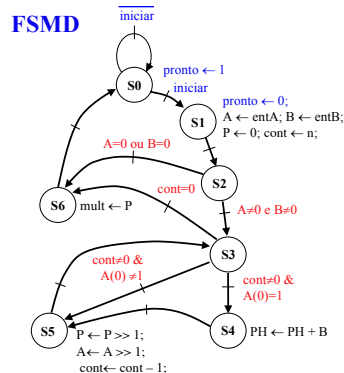
- **n_ciclos** é o nº de ciclos de relógio, no pior caso, para concluir o cálculo
- **T** é o período (mínimo) do relógio

Se $n = 8$ bits:

- Maior inteiro sem sinal: 255 ($\Rightarrow 11111111$)
 - Pior caso: $A=255, B \neq 0$
 - Sequência de execução: S1, S2, 8x[S3,S4,S5], S3, S6
- **28 ciclos de relógio** (BO Mulseq: 768 ciclos, BO Mulseq2: 513 ciclos)

Generalizando para n bits:

- Maior inteiro sem sinal: $2^n - 1$
- Pior caso: $A=2^n - 1, B \neq 0$
- Sequência de execução: S1, S2, $n \times [S3, S4, S5]$, S3, S6 → $3n + 4$ passos ($\approx 3n$ ciclos de relógio) (BO Mulseq: 3×2^n ciclos, BO Mulseq2: $2 \times 2^{n+1}$ ciclos)



Falta estimar T...

Processadores Dedicados

Comparação de Custos: Mulseq x Mulseq2 x Mulseq3

Nome do Bloco Acelerador	Mulseq	Mulseq2	Mulseq3
Característica Principal	Custo mínimo	Desempenho máximo	Algoritmo otimizado
Custo do BO (nº de trans.) n=8 n=16	134n 1072 2144	146n 1168 2336	$172n + 52(1+\log_2 n)$ 1584 3012
Custo do BC nº de estados nº de sinais de controle	6 5	5 4	7 4
Nº de ciclos de relógio n=8 n=16	3×2^n 768 196.608	$2 \times 2^n + 1$ 513 131.073	$3n+4$ 28 52

- Multseq3 tem a vantagem de calcular o resultado com precisão, sem gerar *overflow*
- Multseq e Multseq2 desconsidera a possibilidade de *overflow*.
- Para a comparação de custo do BO ser justa, é necessário alterar os BOs de Multseq e Multseq2, de modo que eles consigam realizar o cálculo com precisão, assim como Multseq3 o faz

Processadores Dedicados

Comparação de Custos: Mulseq x Mulseq2 x Mulseq3

Nome do Bloco Acelerador	Mulseq	Mulseq2	Mulseq3
Característica Principal	Custo mínimo	Desempenho máximo	Algoritmo otimizado
Custo do BO (nº de trans.) n=8 n=16	134n 1072 2144	146n 1168 2336	$172n + 52(1+\log_2 n)$ 1584 3012
Custo do BC nº de estados nº de sinais de controle	6 5	5 4	7 4
Nº de ciclos de relógio n=8 n=16	3×2^n 768 196.608	$2 \times 2^n + 1$ 513 131.073	$3n+4$ 28 52

Conclusão principal:

Em geral, alterações no algoritmo tem impacto maior no custo e no desempenho do que alterações na arquitetura do bloco operativo.

Processadores Dedicados

Comparação de Custos: Mulseq x Mulseq2 x Mulseq3

Tarefas de Casa:

- 1) Usando as características de atraso propostas na aula 3T (tabela de atrasos), estimar o período do relógio e o tempo de execução (TE) de cada uma das 3 versões de Mulseq.
- 2) Faça o Exercício 6 da 2ª Lista de Exercícios*
- 3) Faça o Exercício 7 da 2ª Lista de Exercícios*

* Disponível no topo da página Moodle da disciplina.