

Projeto Sistemas Digitais

- Projeto de Sistemas Digitais: modelagem do algoritmo por fluxograma
- Implementações para Parte de Controle e Parte Operativa

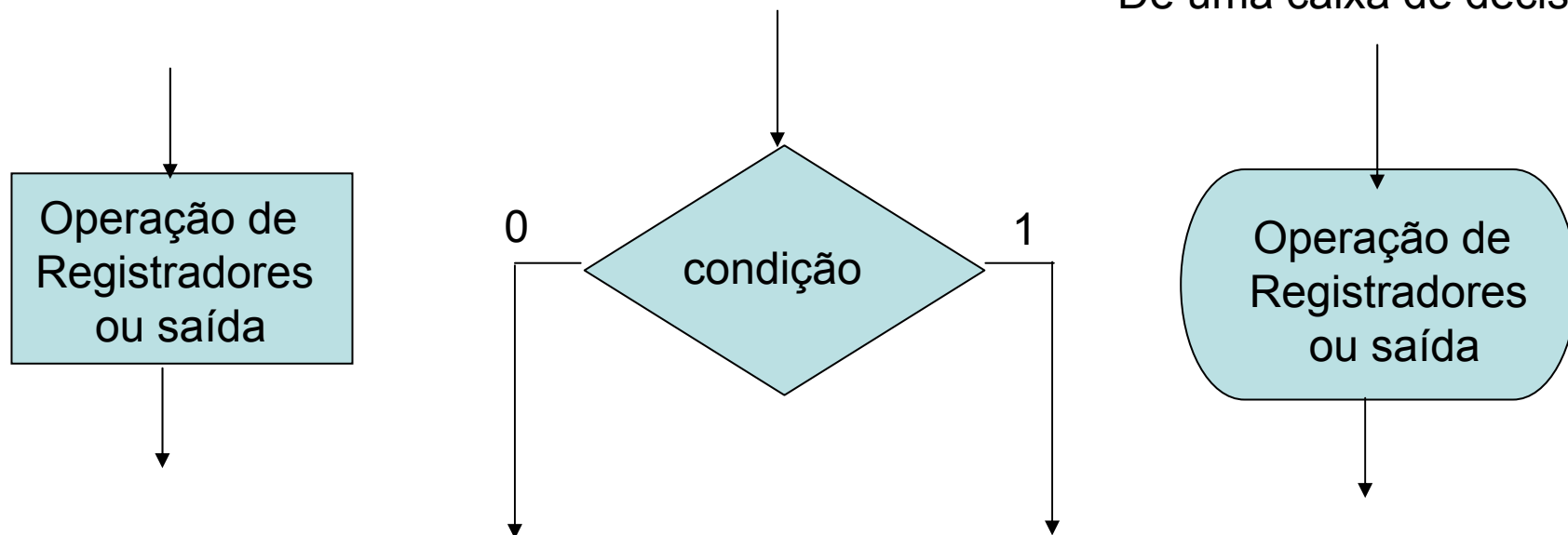
Modelagem: Algorithmic State Machine (ASM)

Aula
2

- Um fluxograma é um modo conveniente de especificar uma sequência de passos procedurais e caminhos de decisão para um algoritmo.
- Fluxograma para um algoritmo implementado em hardware necessita características especiais que conecta ao hardware que implementa o algoritmo.
- Esse fluxograma especial é chamado de **Algorithmic State Machine (ASM)**, onde *state machine* é usado aqui para denominar um circuito sequencial.

Um fluxograma ASM descreve uma **sequencia de eventos** e a relação de **tempo** entre os estados da unidade de controle e as **ações** que ocorrem em cada estado em resposta aos pulsos de relógio.

- O fluxograma ASM contém 3 elementos básicos:
 - Caixa de estado
 - Caixa de decisão
 - Caixa de saída condicional

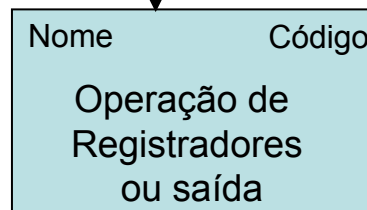


Algorithmic State Machine (ASM)

Aula
2

Caixa de estado

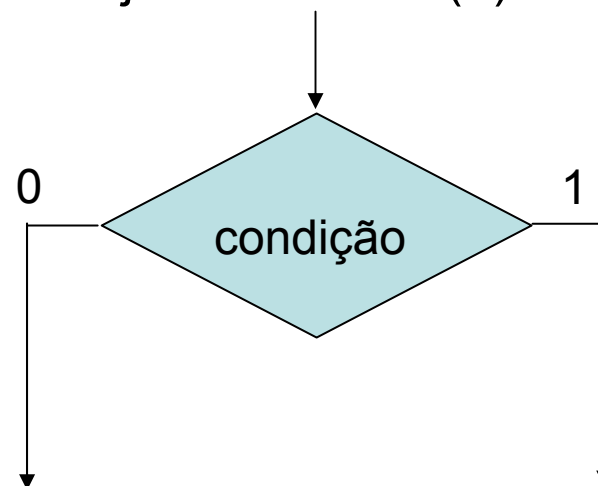
- A caixa de estado contém:
 - **operações de transferência** entre registradores ou **sinais de saída que são ativados** enquanto a unidade de controle está neste estado.
- O nome simbólico do estado é colocado no canto superior esquerdo normalmente e o código binário do estado (quando for definido) é colocado no canto superior direito.



Algorithmic State Machine (ASM)

Caixa de decisão

- A caixa de decisão descreve o efeito da entrada no controle.
- Possui dois caminhos de saída.
- A condição de entrada é uma única variável ou uma expressão booleana nesta caixa.
- Um caminho é escolhido caso a condição seja verdadeira (1) e o outro caso ela seja falsa (0).

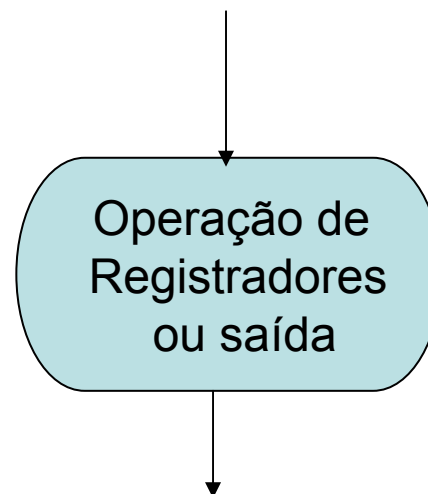


Algorithmic State Machine (ASM)

Caixa de saída condicional

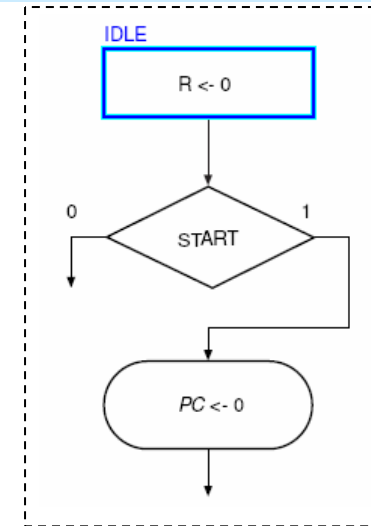
- A caixa de saída condicional é única do fluxograma ASM e sua entrada deve vir de uma ou mais caixas de decisão.
- Se a condição especificada em um caminho que passa pela caixa de decisão vindo de uma caixa de estado é satisfeito, a transferência de registradores ou saída listada na caixa de saída condicional é ativada.

De uma caixa de decisão

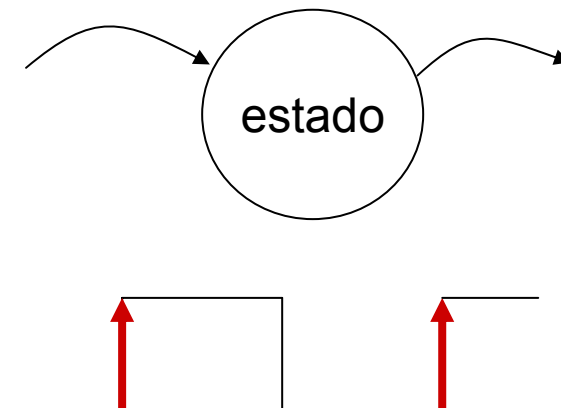


Definição de Bloco ASM

- Um **bloco ASM** consiste de uma caixa de estado e todas as caixas de decisão e saídas condicionais conectadas entre a saída da caixa de estado e os caminhos de entrada para a mesma ou outras caixas de estado.
- O **bloco** representa **decisões e ações** de saída que acontecem no estado.
- Qualquer saída cuja condição é satisfeita no bloco ASM tem sua ativação no estado em questão.
- Qualquer transferência entre registradores, que acontece quando condições são satisfeitas, **são realizadas quando o evento do clock acontece**.
- Este mesmo evento do clock também transfere o controle para um dos próximos estados, como especificado pelas decisões do bloco ASM.

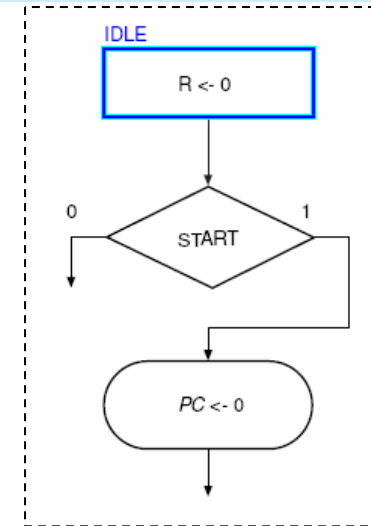


Equivale a...

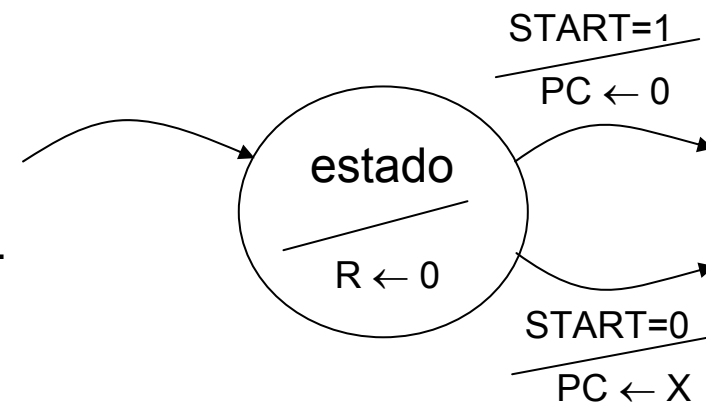


Definição de Bloco ASM

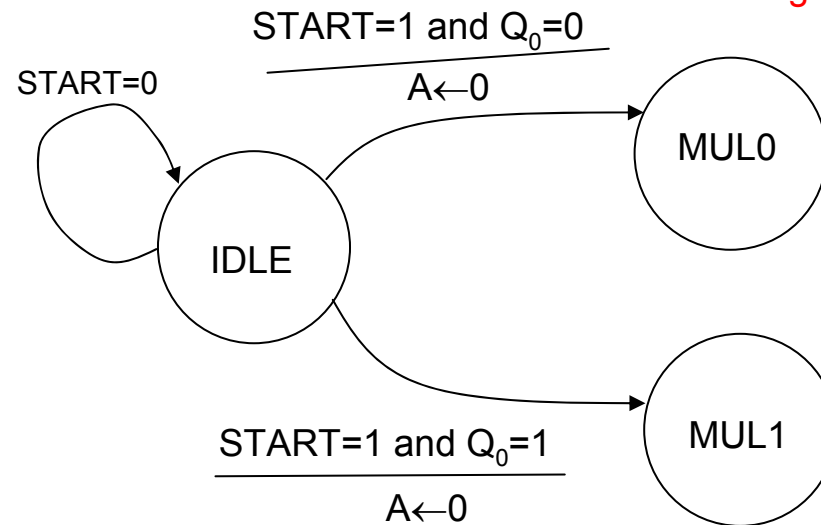
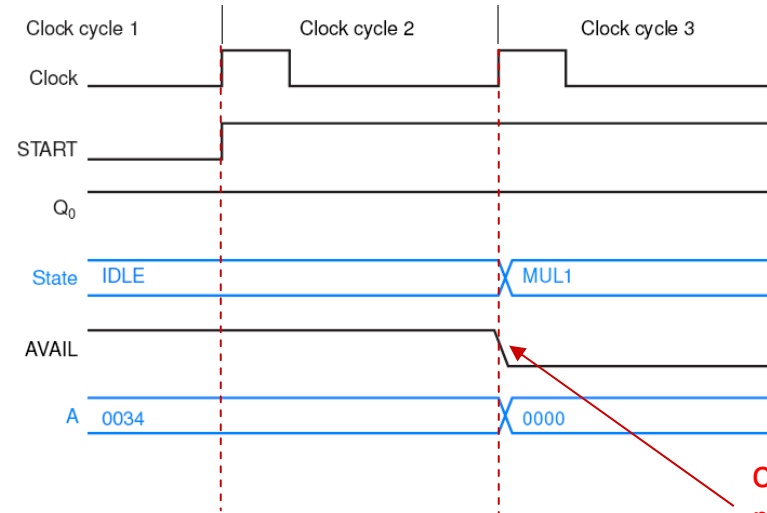
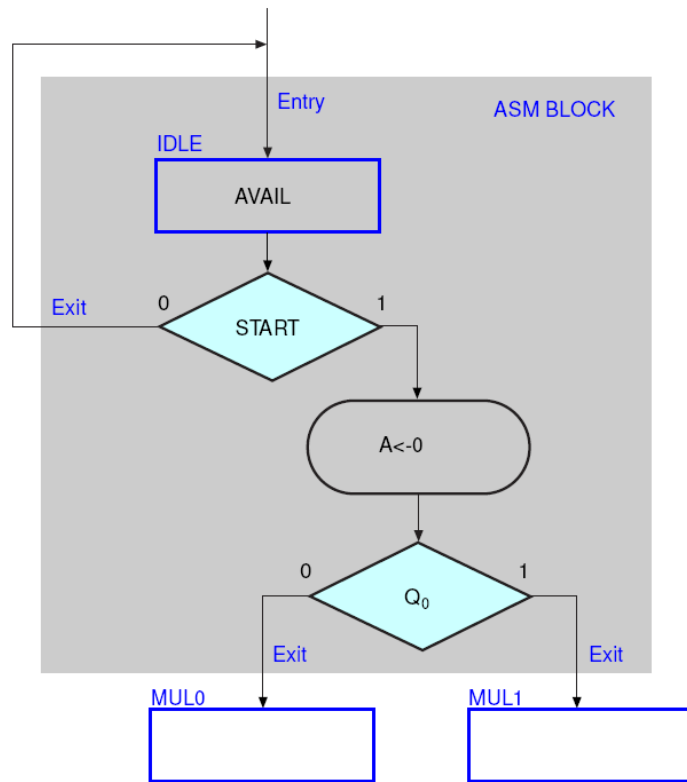
- O **fluxograma ASM** é uma forma eficiente de representar diagramas de estados da parte de controle de circuitos sequenciais.
- Cada bloco ASM corresponde a um nodo do diagrama de estados.
- As **caixas de decisão** são equivalentes aos valores de entrada descritos nos arcos que conectam os estados do diagrama.
- As transferências entre registradores e saídas nas **caixas de estado** são as saídas definidas nos estados do diagrama de estados.
- As saídas nas **caixas de decisão** são as saídas condicionais descritas nos arcos que conectam os estados no diagrama de estados.



Equivale a...

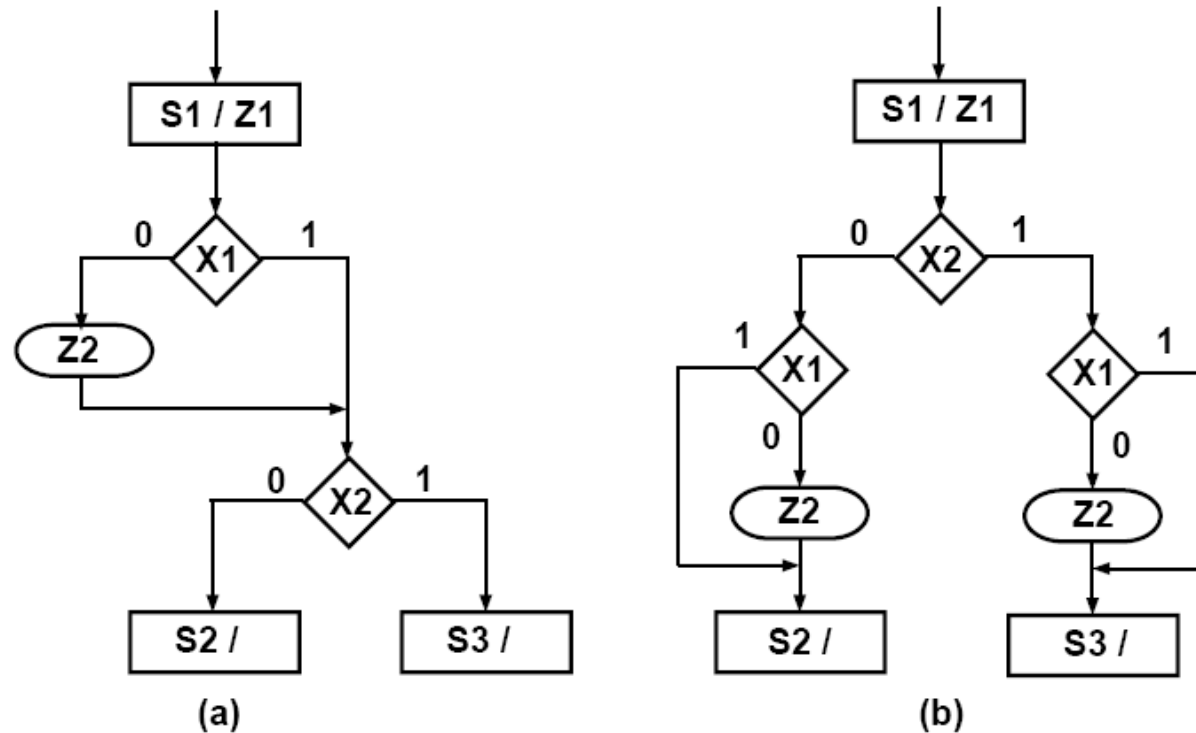


Exemplo de um bloco ASM



Exemplo de Blocos ASM equivalentes

Aula
2

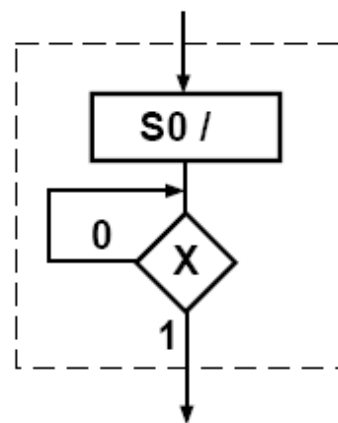


Sabendo que:

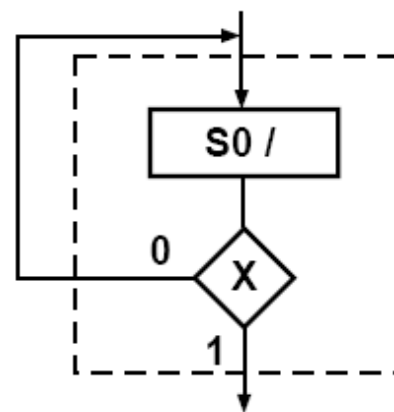
- S1, S2 e S3 são nomes simbólicos dos estados
- X1 e X2 são entradas
- Z1 e Z2 são saídas

Exemplo de Erro em Laço

Aula
2



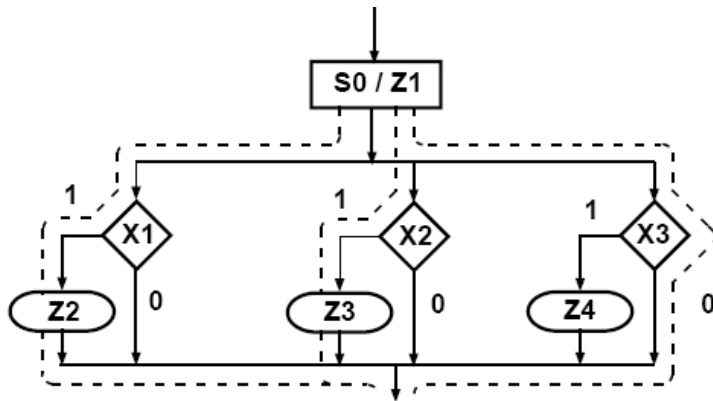
Incorreto



Correto

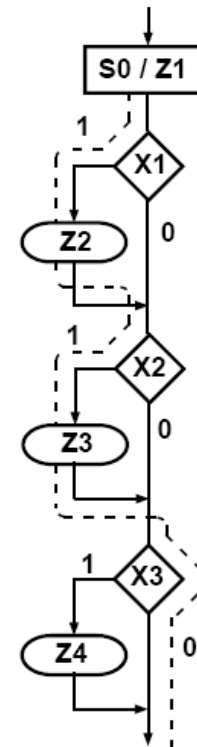
Exemplo de Blocos ASM equivalentes

PARALELO



Equivale a 1 estado

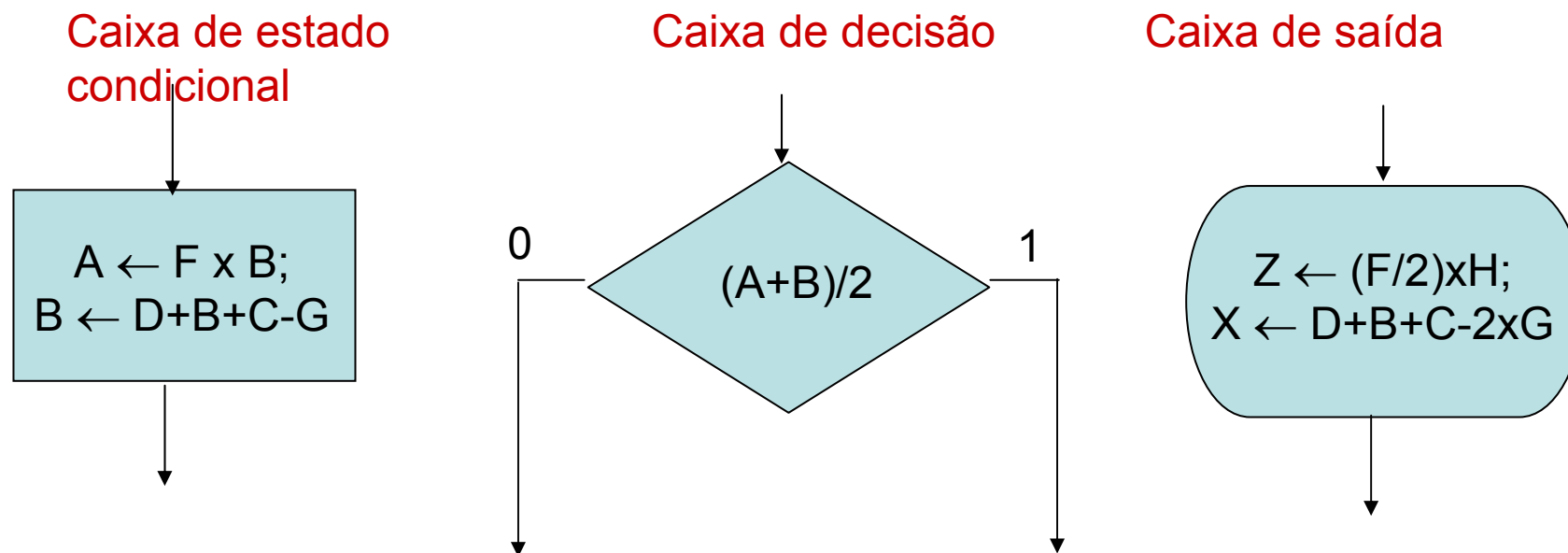
SERIAL



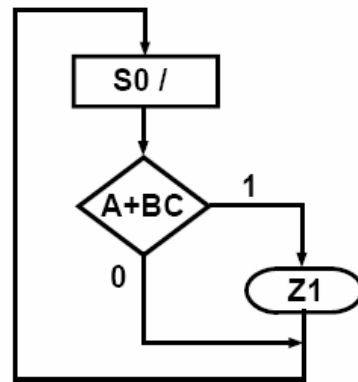
Equivale a 1 estado

... Podemos definir operações mais complexas nas atribuições que serão implementadas por operadores (somadores, multiplicadores, etc...) fora da máquina de estados (parte de controle) e sim em uma parte operativa.

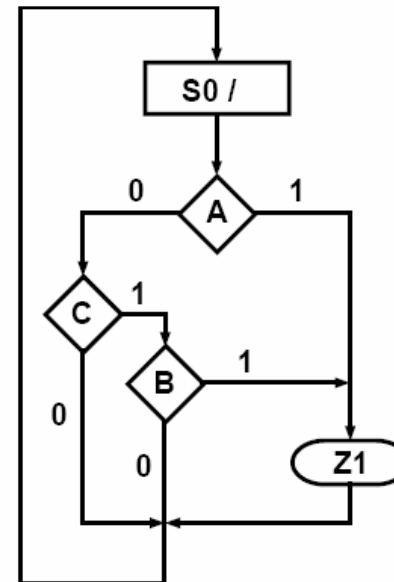
Exemplos:



Exemplo de Implementações



(a)



(b)

Precisa OPERADORES de:

- and
- or

A+BC

Não precisa de OPERADORES
• porem essa lógica de decisão é implementada na função de próximo estado na FSM

OBJETIVO

- De um ASM chegar ao circuito final (parte operativa e parte de controle)

Exemplo - ASM:rami

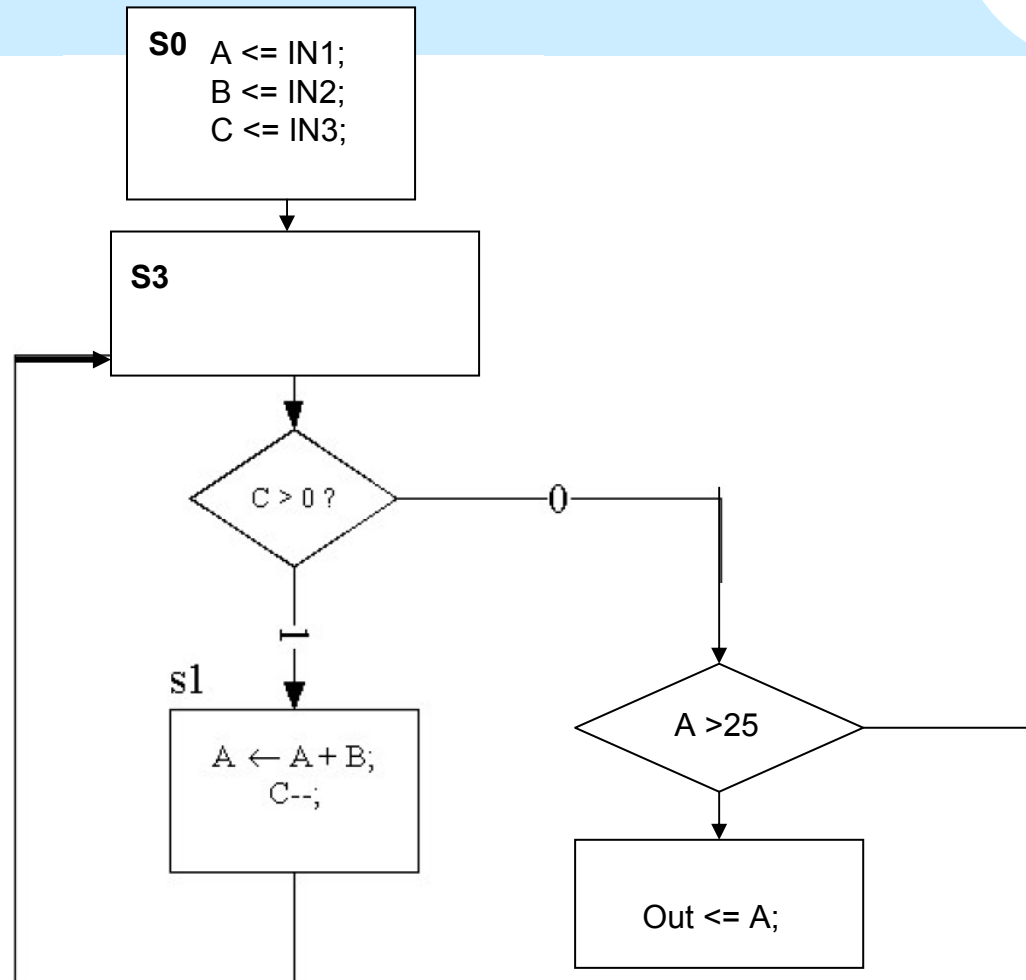
```
s0
  A ← IN1;
  B ← IN2;
  C ← IN3;

c0
  If (C > 0) (goto s1);
  Else (goto s2);

s1
  A ← A + B;
  C ← C - 1;
  (goto c0);

s2
  If (A > 25) Out ← A;
  (goto s2);
```

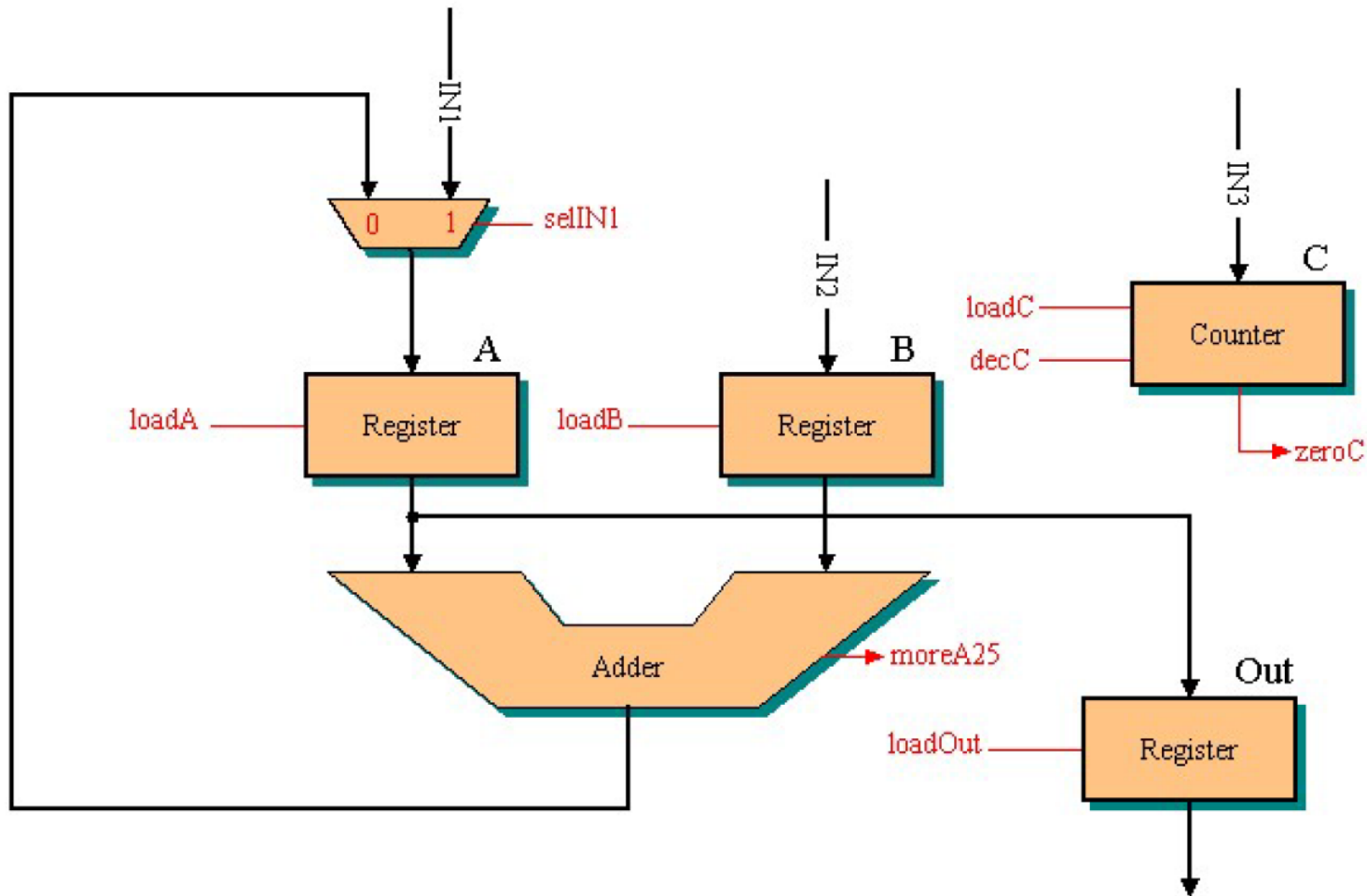
Textual ASM Chart



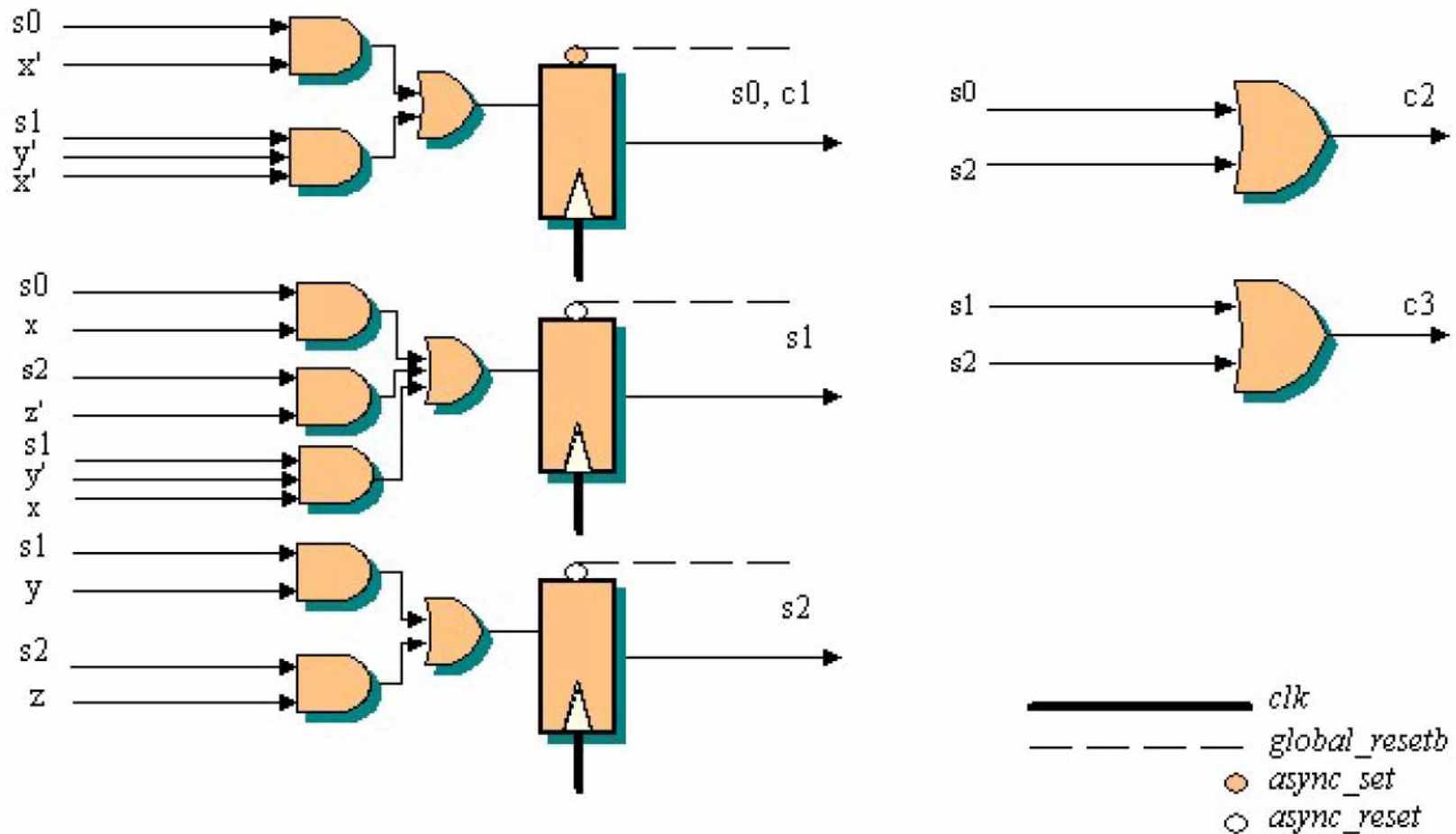
Graphical ASM Chart

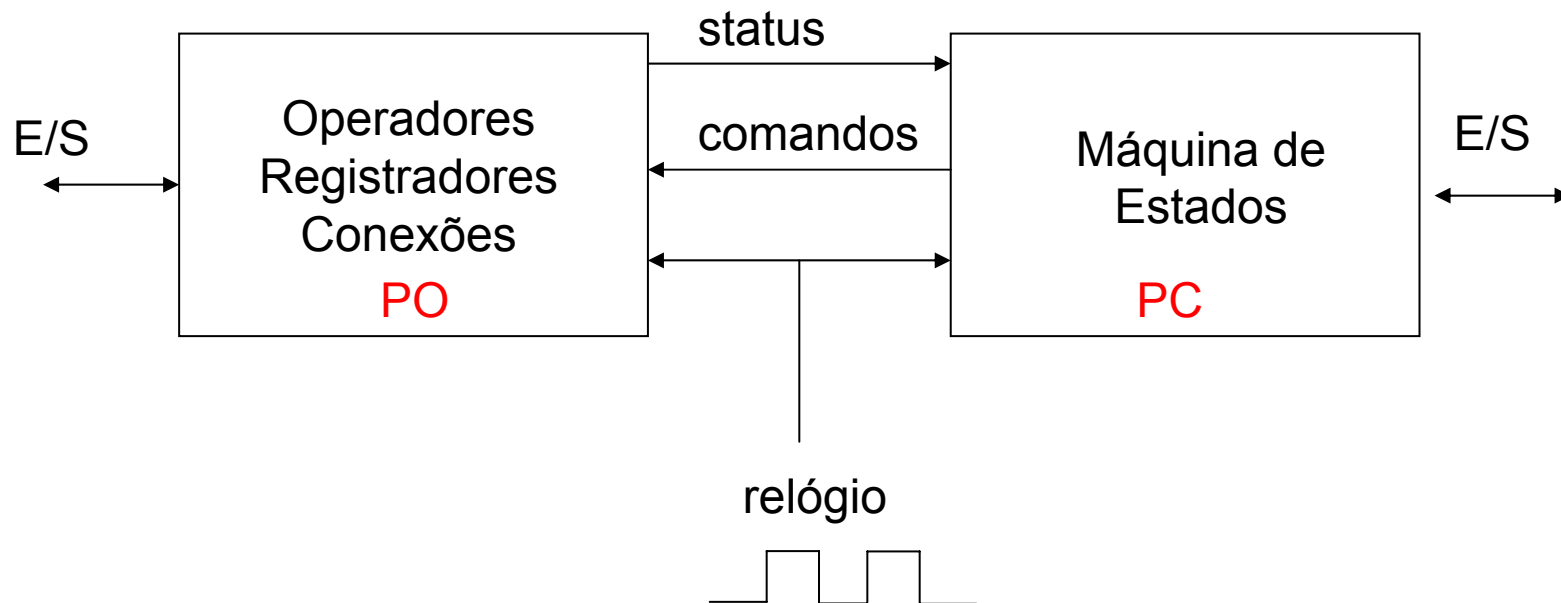
Parte Operativa (Datapath) do ASM:rami

Aula
2

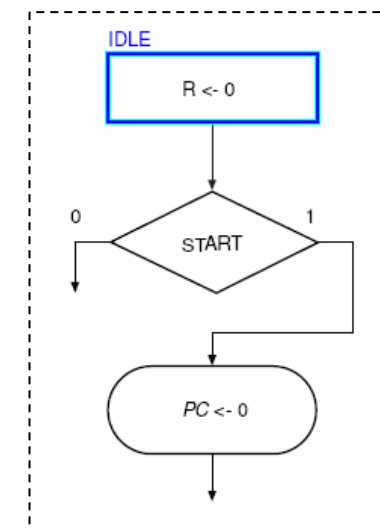


Parte de Controle do ASM:rami



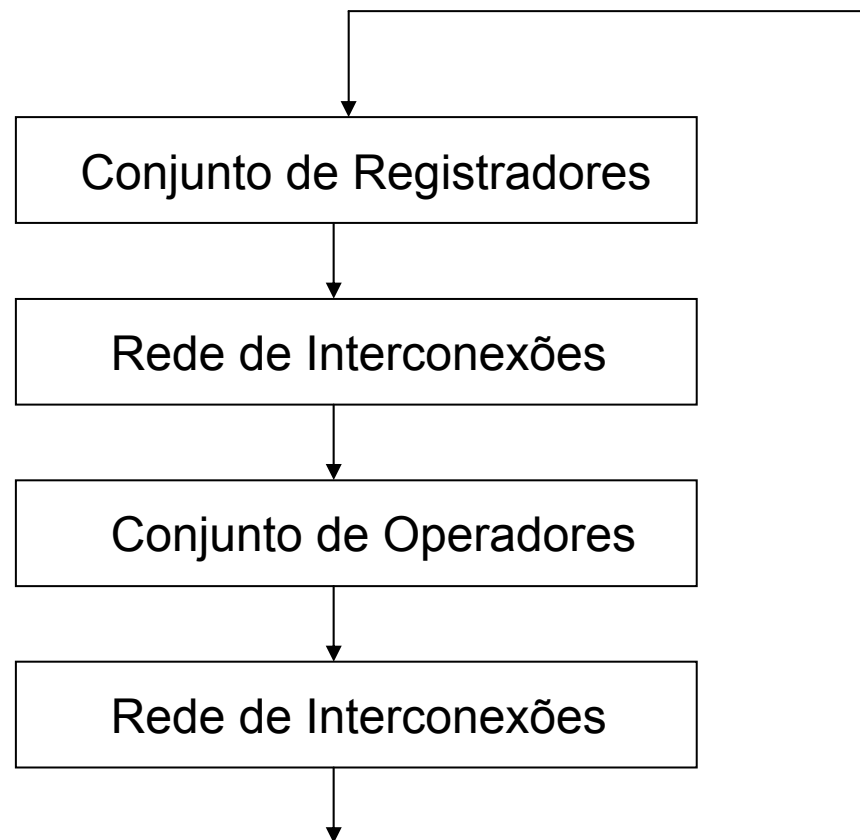


- A partir do fluxograma pode-se obter um bloco de controle onde cada bloco ASM do fluxograma corresponderá a um estado no controle. (desperdício x técnicas de otimização)
- Fluxograma possui operações factíveis do ponto de vista de sistemas digitais.
- Operações que podem ser realizadas em um ciclo de relógio.
- Decisão dos recursos disponíveis na parte operativa, interdependencia com o projeto da parte de controle.



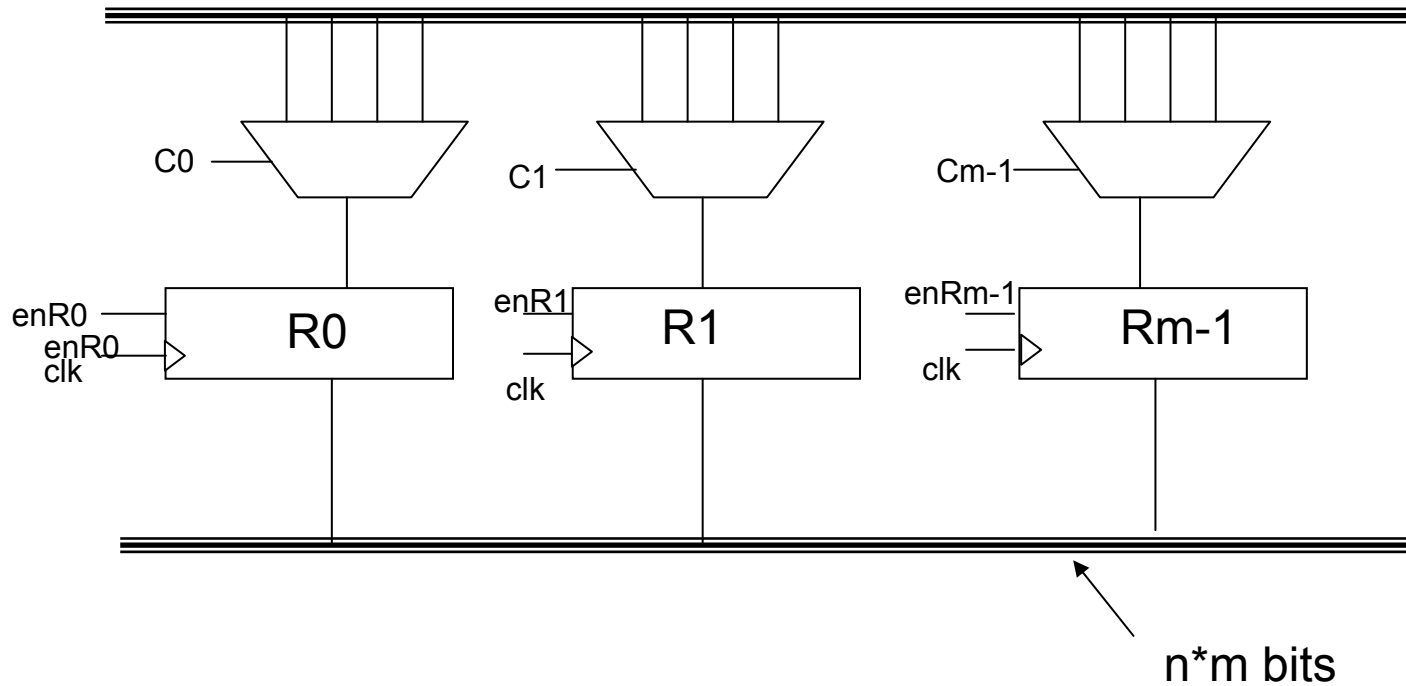
Custo das conexões

- Durante a simulação (Matlab, C, etc...), o custo de uma ligação entre uma variável e outra é gratuita. Isto não reflete a realidade dos circuitos.
- A temporização ou o atraso dos componentes envolvidos no projeto está escondida quando se usa linguagem de alto nível.



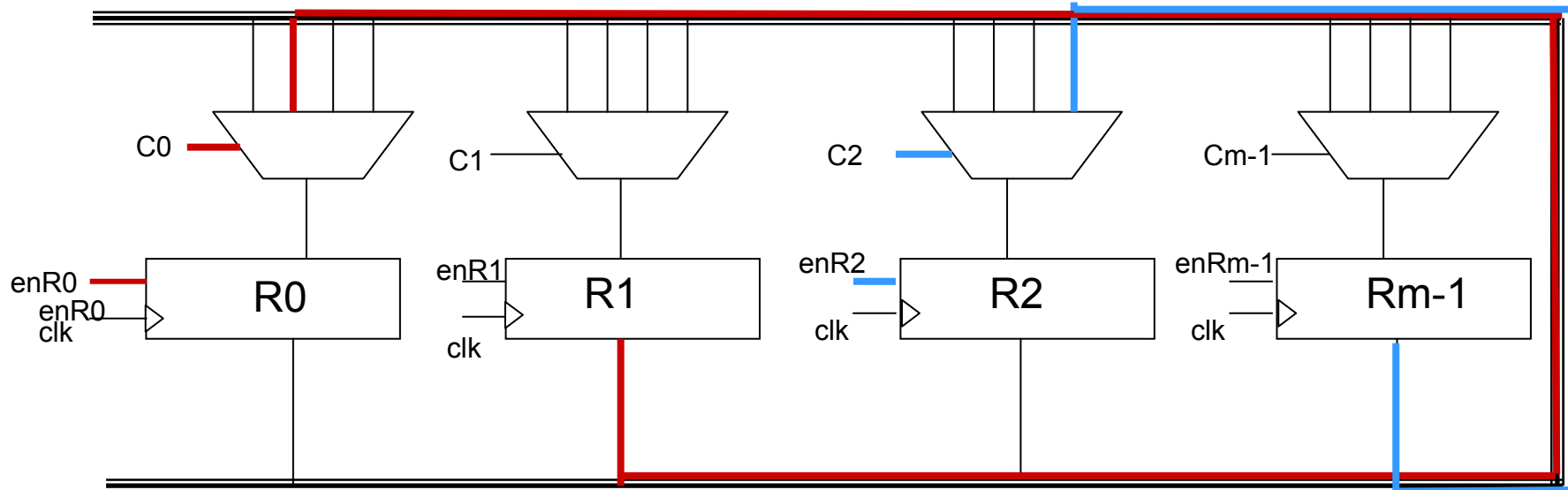
- m registradores de n-bits cada
- Cada registrador tem um mux de m:1 de n-bits por entrada
- Se C_m é o custo de 1 mux 2:1

Custo total = $m * C_m * (n-1)$



Transferências simultaneas = + paralelismo

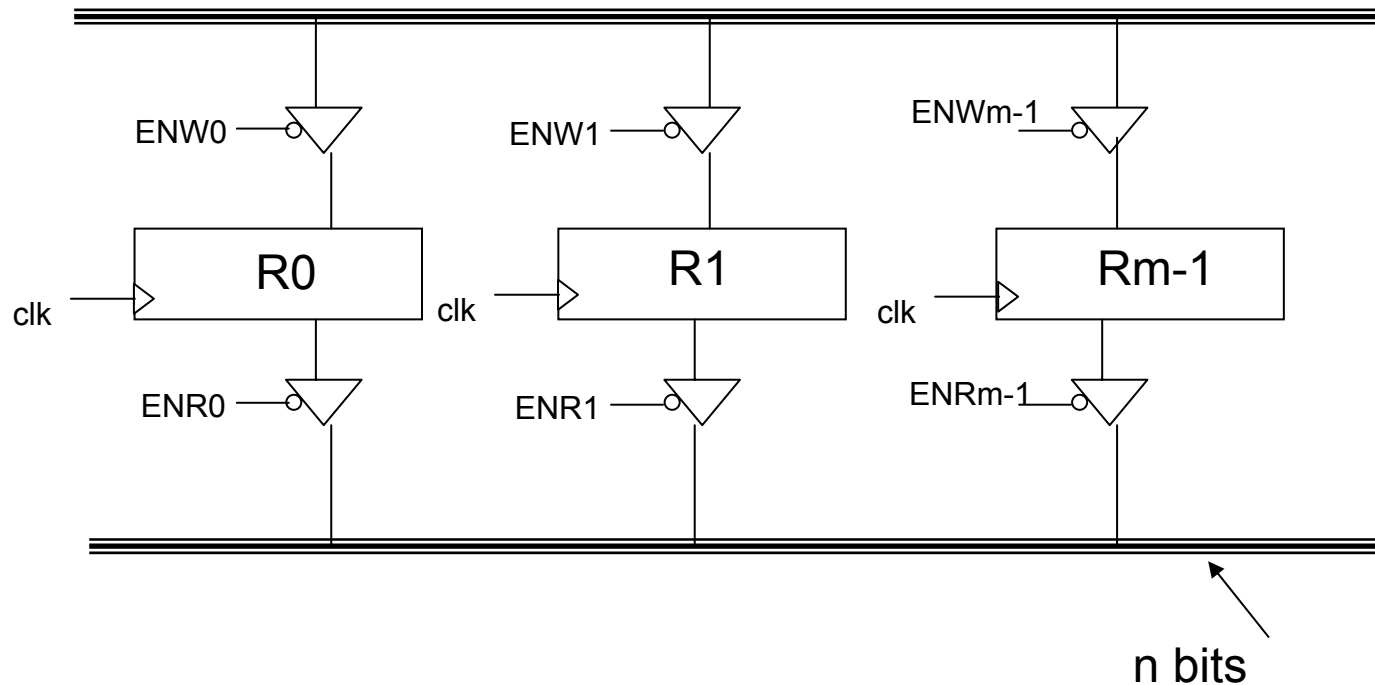
$R0 \leftarrow R1$ ao mesmo tempo que $R2 \leftarrow Rm-1$



Conexões com Barramento

- m registradores de n-bits cada
- Se C_b é o custo do buffer de 3 estados

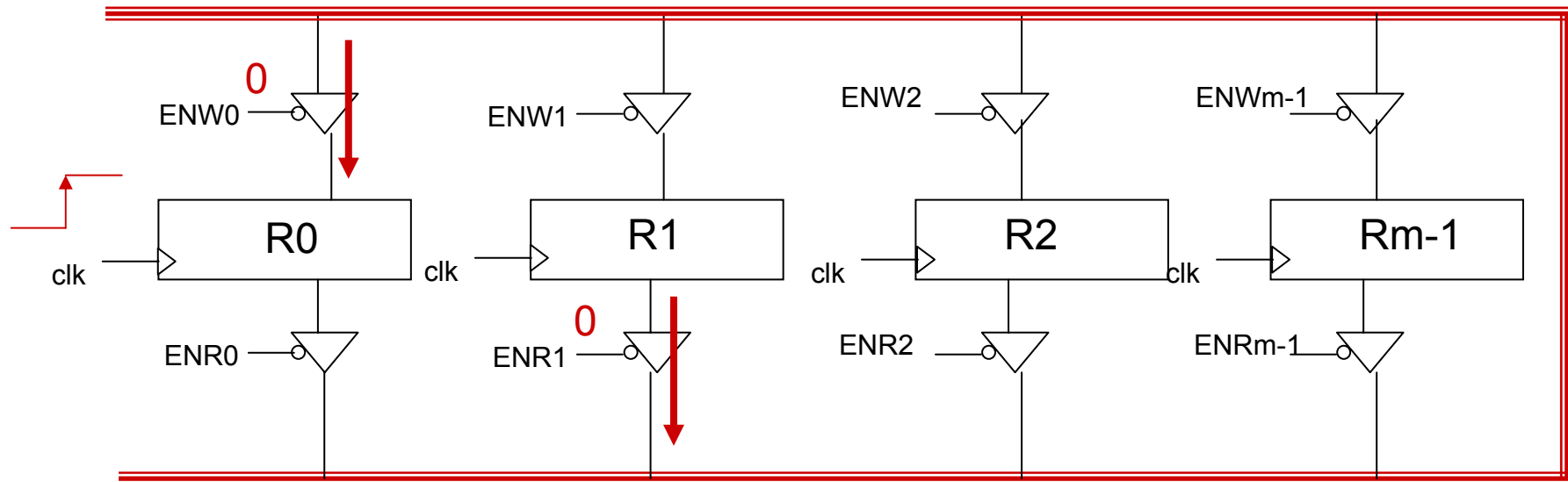
Custo total = $m * C_b * n$ ou $2 * m * C_b * n$



Serialização das transferências

Exemplo: $R0 \leftarrow R1$

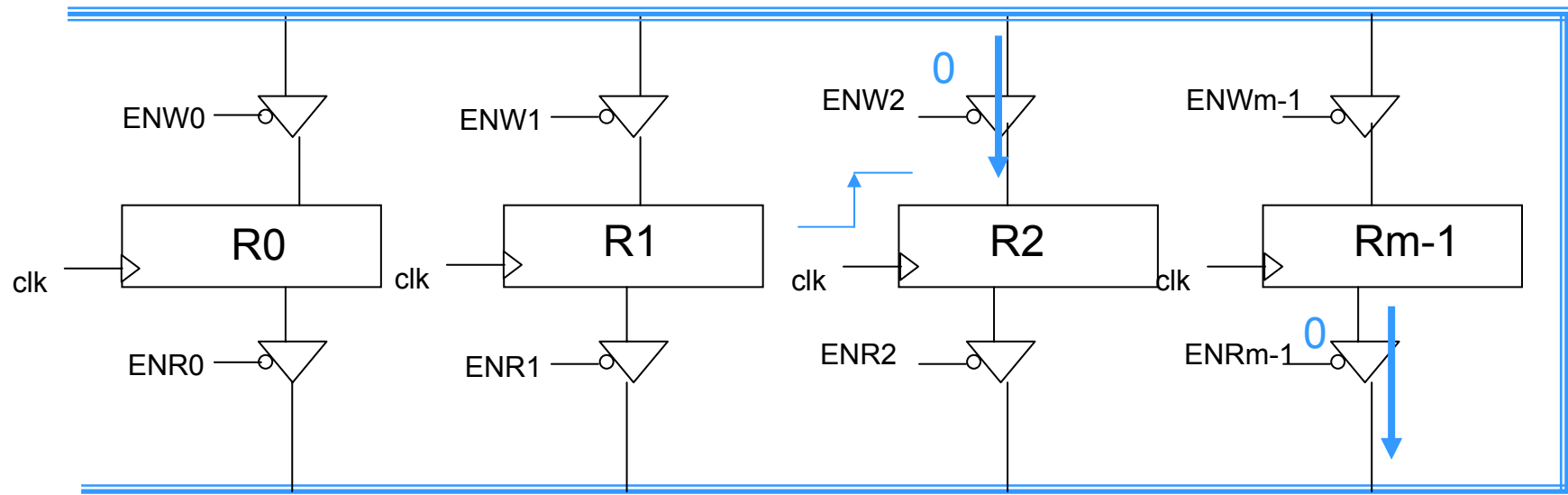
ciclo de clock 1



Serialização das transferências

Exemplo: $R2 \leftarrow R_{m-1}$

ciclo de clock 2



Custo mux para inserção de registradores

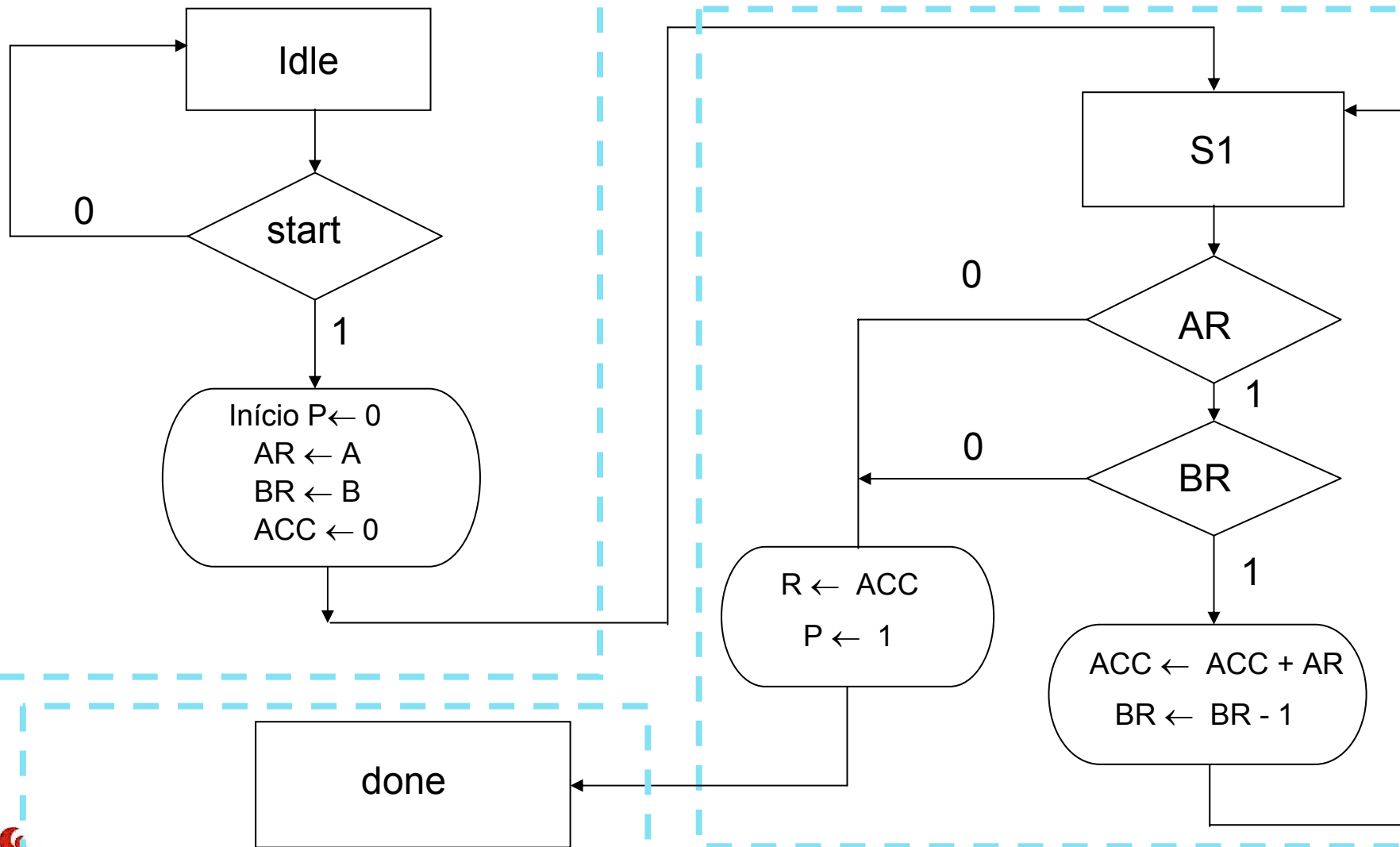
$(m+1) * C_m * (n-1+m)$: quadrático em termos de m

Custo barramento para inserção de registradores

$(m+1)*C_b*n$

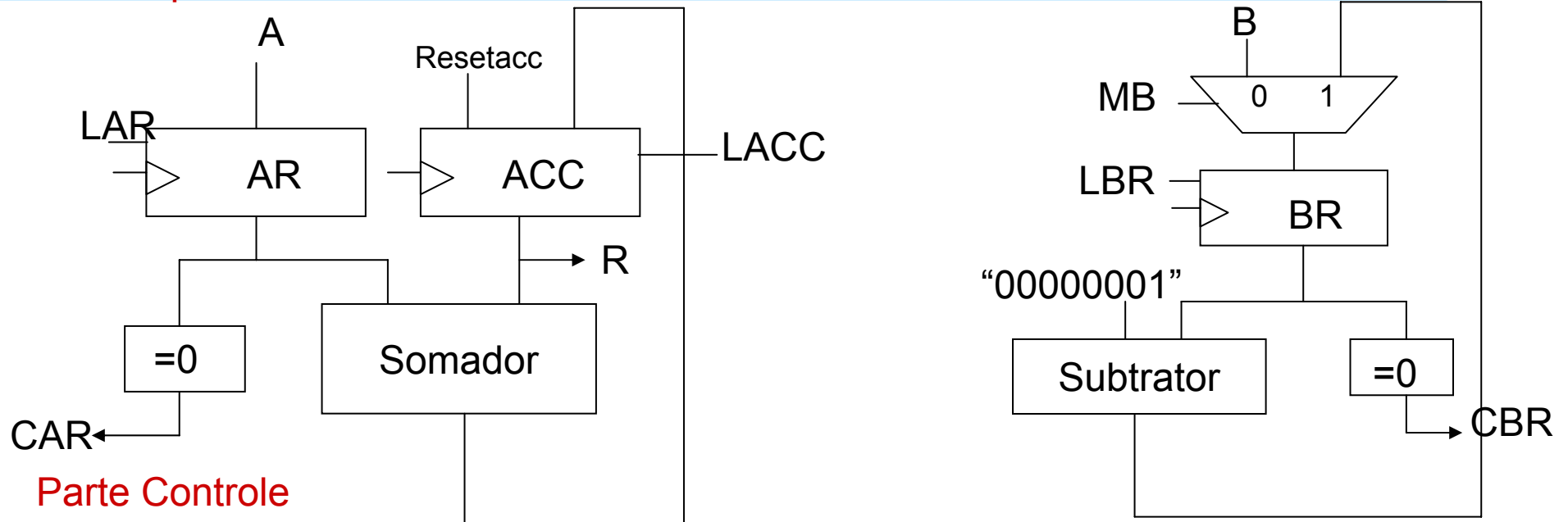
- Determinar os recursos a serem disponíveis na PO para determinar o paralelismo das operações.
1. Montagem da tabela de transferência: tabela onde cada variável parece em uma coluna e as linhas são ordenadas pelo estado. Todas as operações que mudam a variável naquele estado são registradas.
 2. Alocação de variáveis em registradores. Determinar quantas variáveis irão ser alocadas em registradores tentando minimizar o número de registradores diferentes. (register allocation and life-time analysis)
 3. Escolha dos operadores necessários

Exemplo: Multiplicador por somas sucessivas

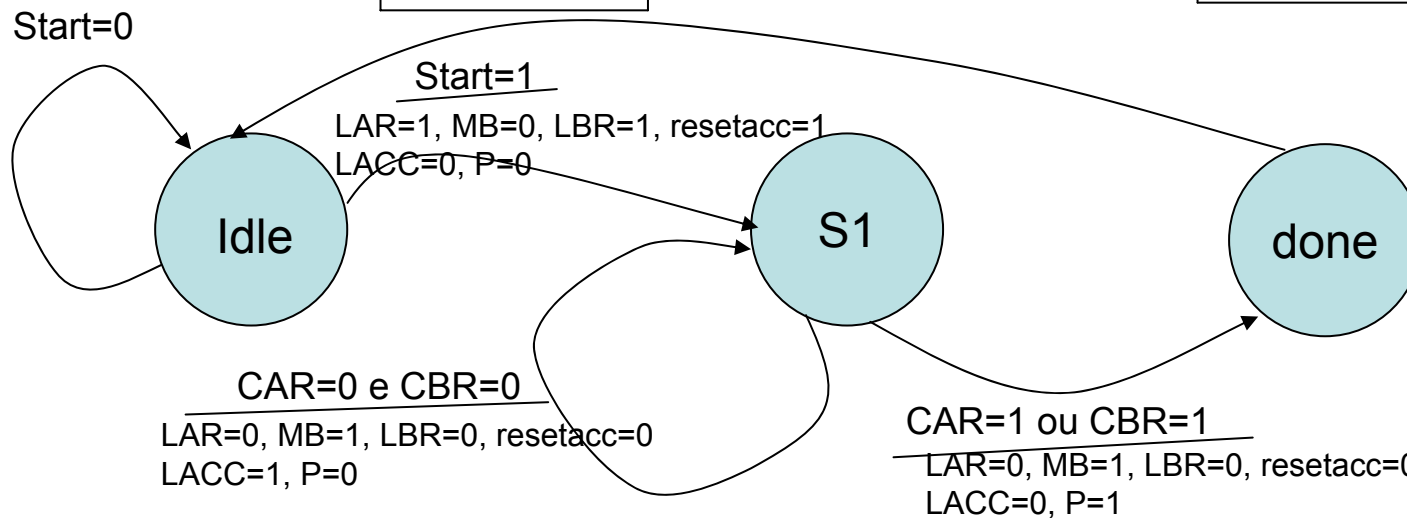


Exemplo: Multiplicador por somas sucessivas

Parte Operativa



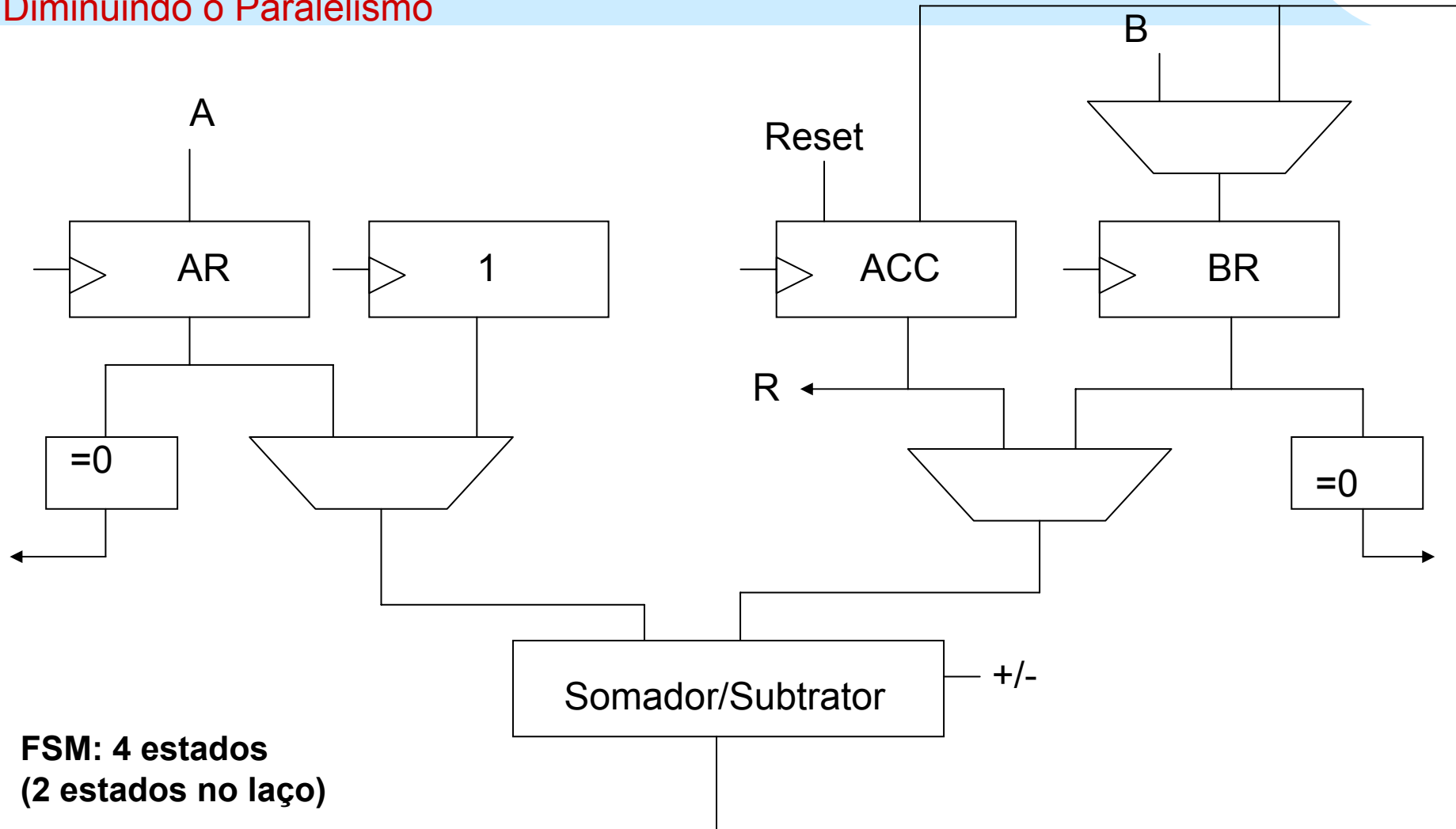
Parte Controle



PO com operadores e mux p/ multiplicação

Aula
2

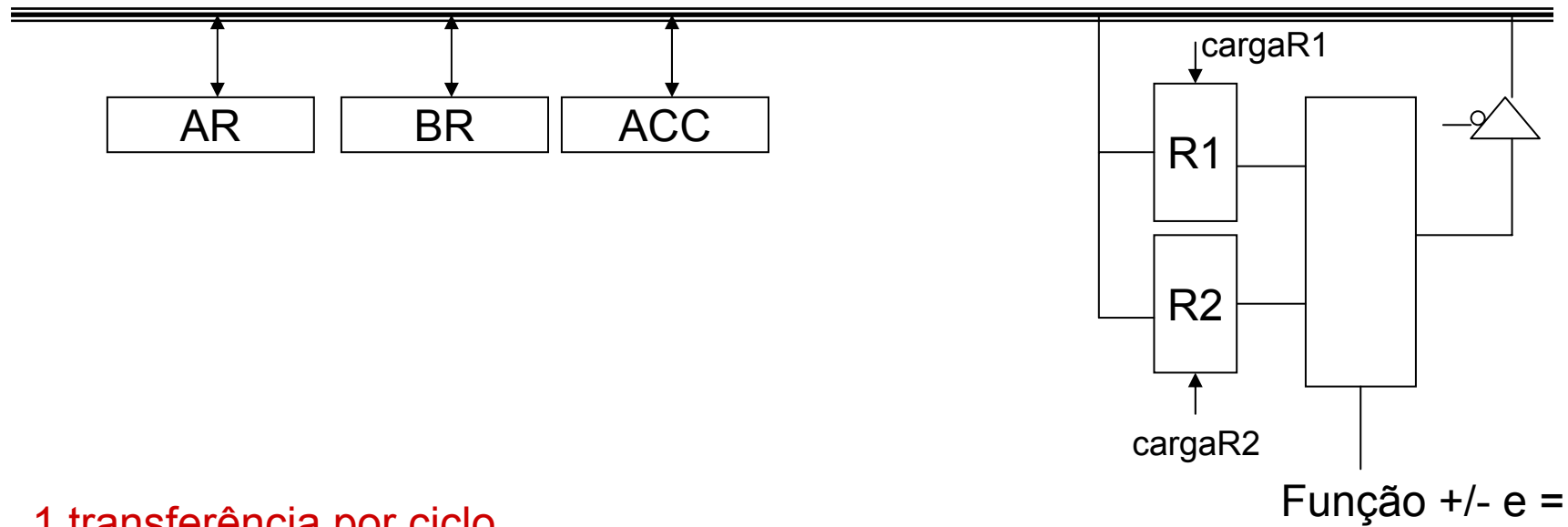
Diminuindo o Paralelismo



PO com Barramento p/ multiplicação

Aula
2

E/S

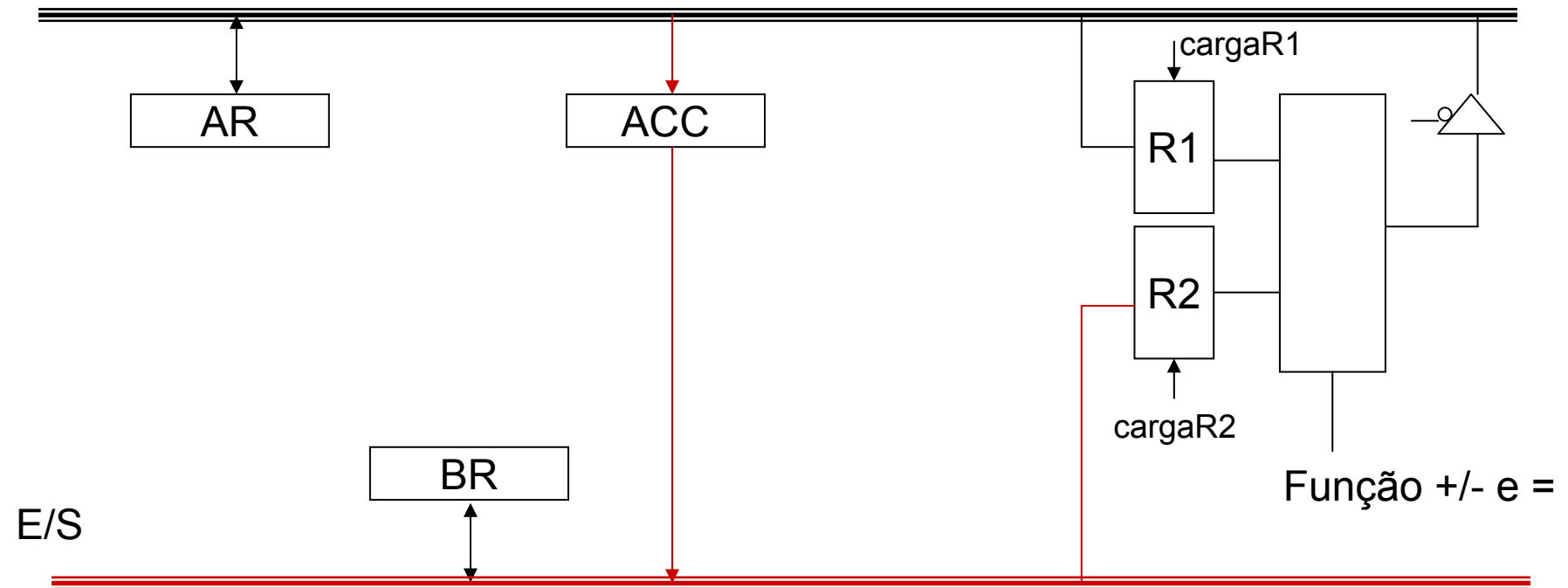


1 transferência por ciclo

PO com Barramento p/ multiplicação

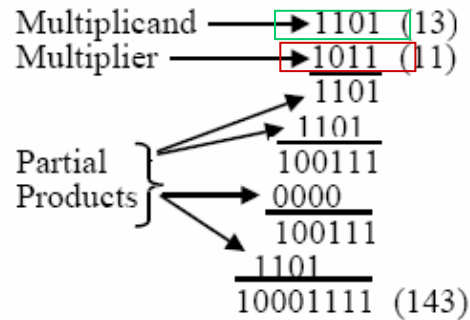
Aula
2

E/S



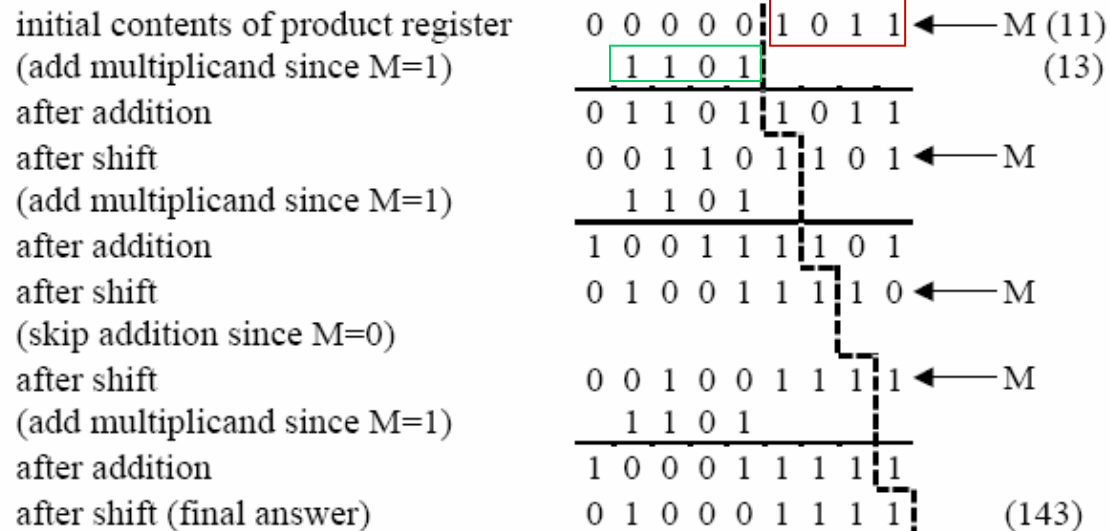
2 transferências por ciclo

Exemplo: Multiplicador Binário



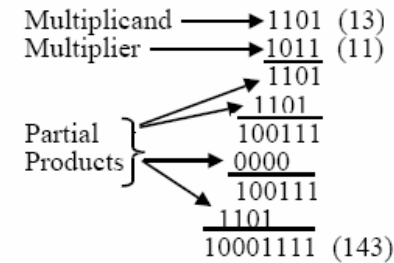
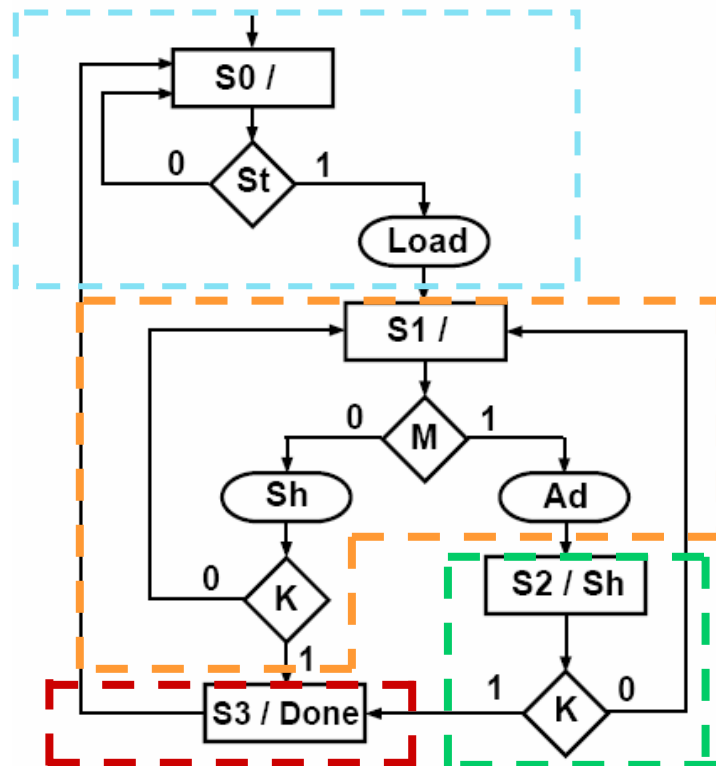
Variável
K

- 1 shift
- 2 shift
- 3 shift
- 4 shift

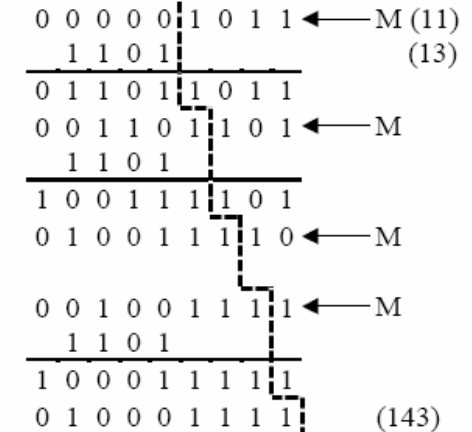


dividing line between product and multiplier

Exemplo: Multiplicador Binário



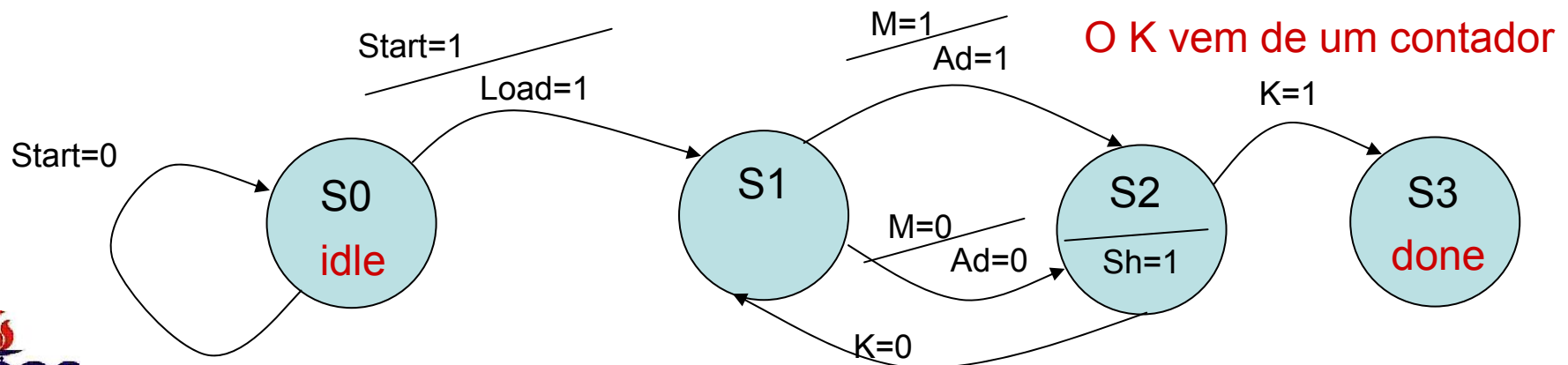
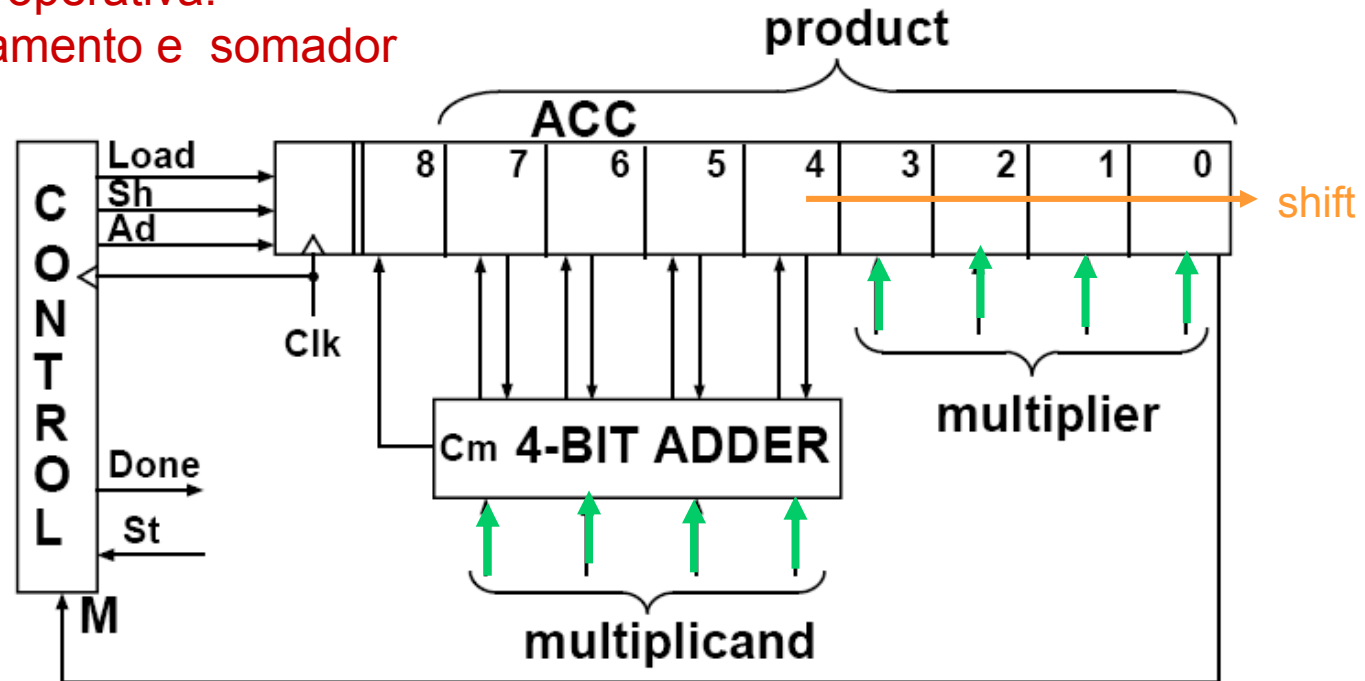
initial contents of product register
 (add multiplicand since M=1)
 after addition
 after shift
 (add multiplicand since M=1)
 after addition
 after shift
 (skip addition since M=0)
 after shift
 (add multiplicand since M=1)
 after addition
 after shift (final answer)

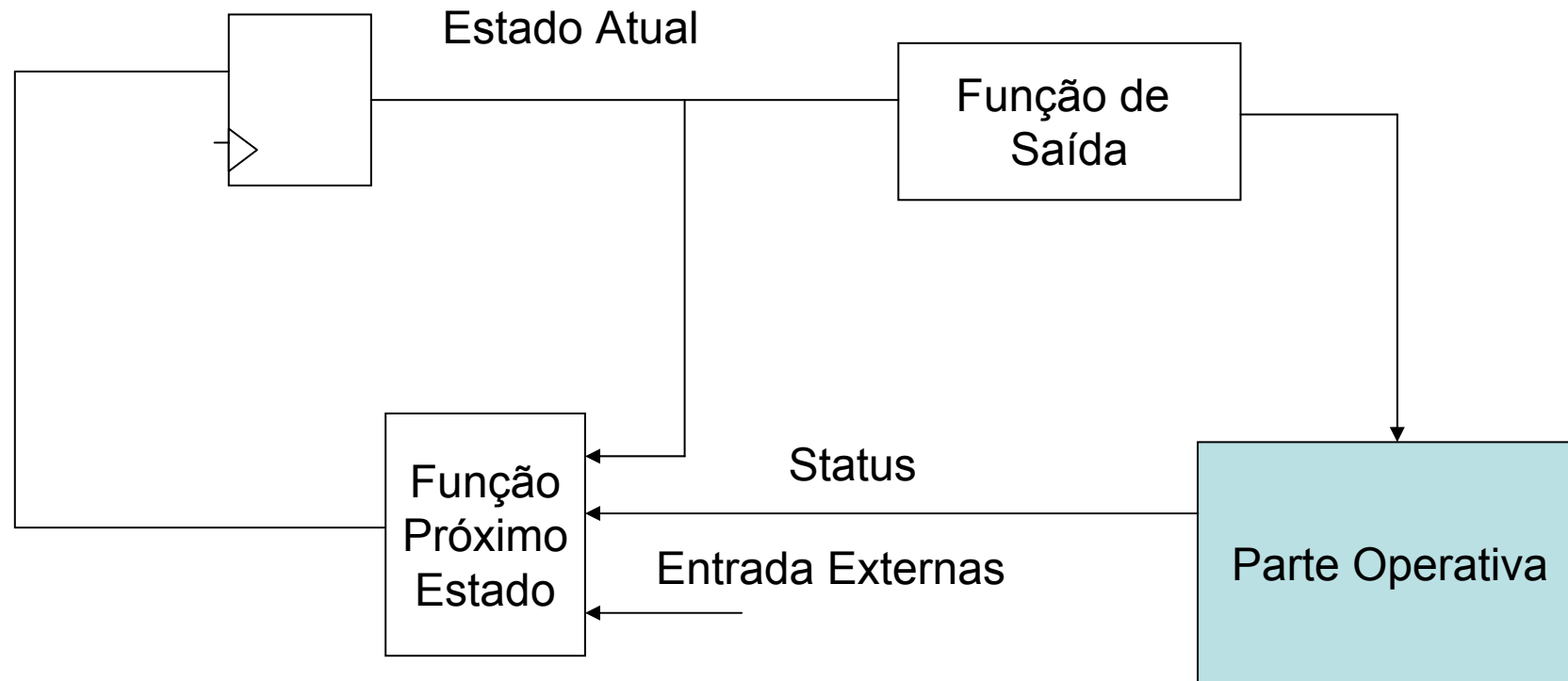


dividing line between product and multiplier

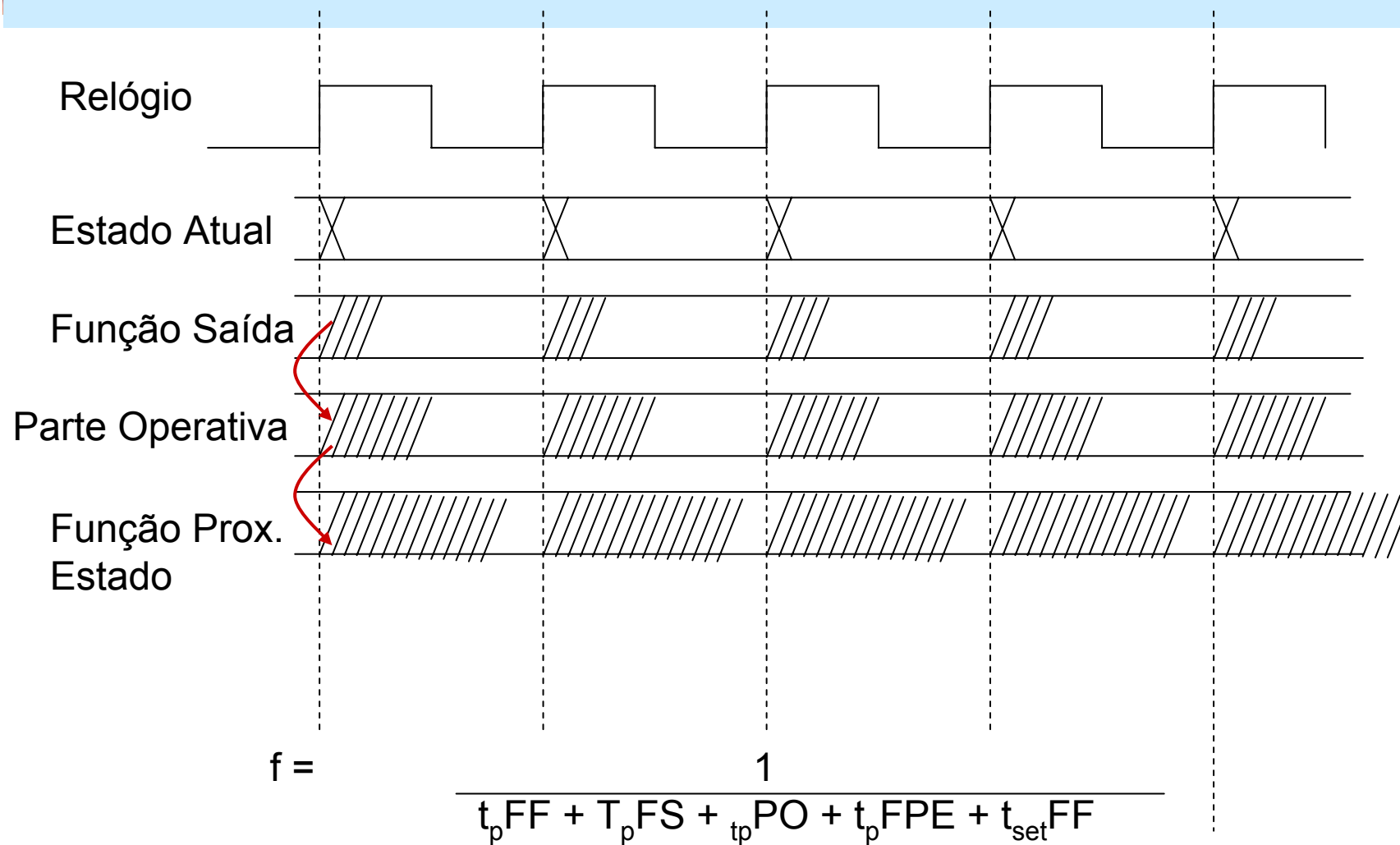
Exemplo: Multiplicador Binário

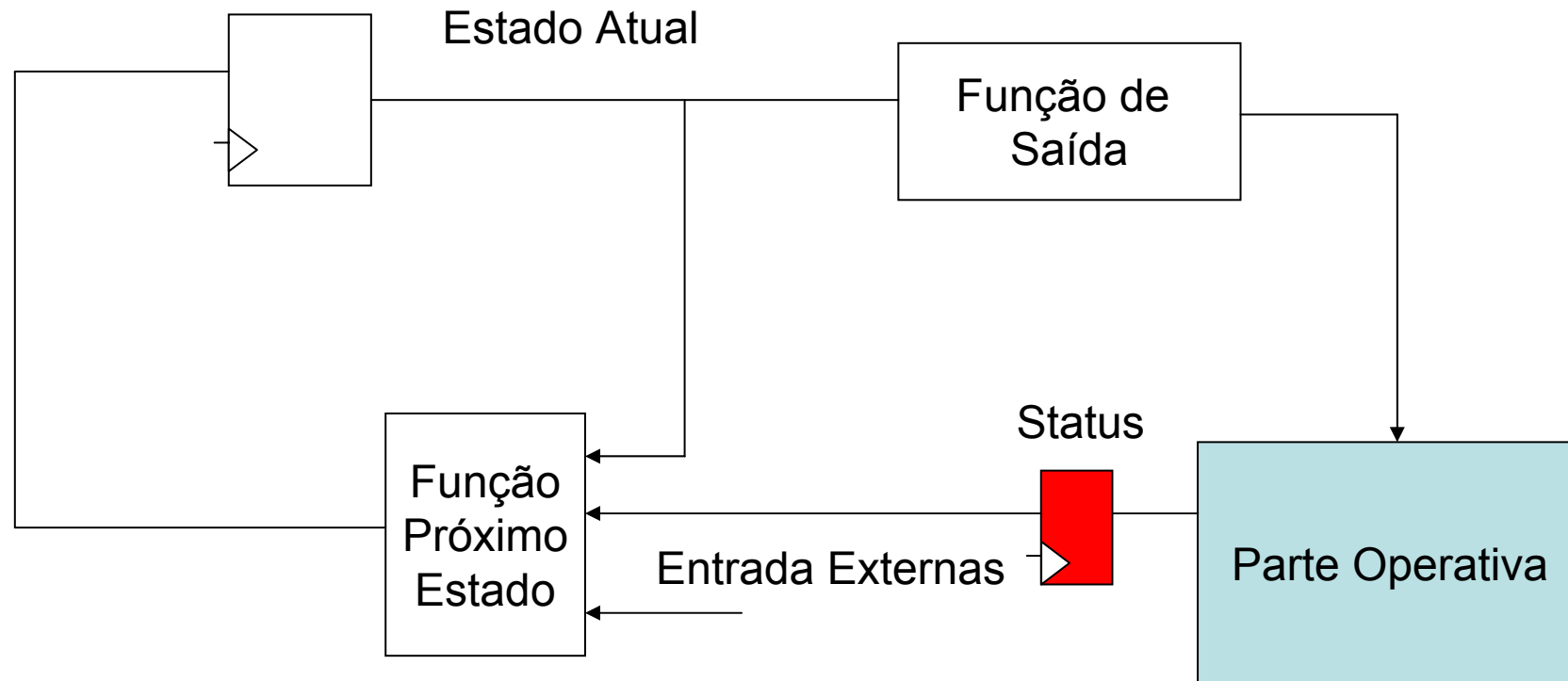
Recursos da parte operativa:
registrador deslocamento e somador



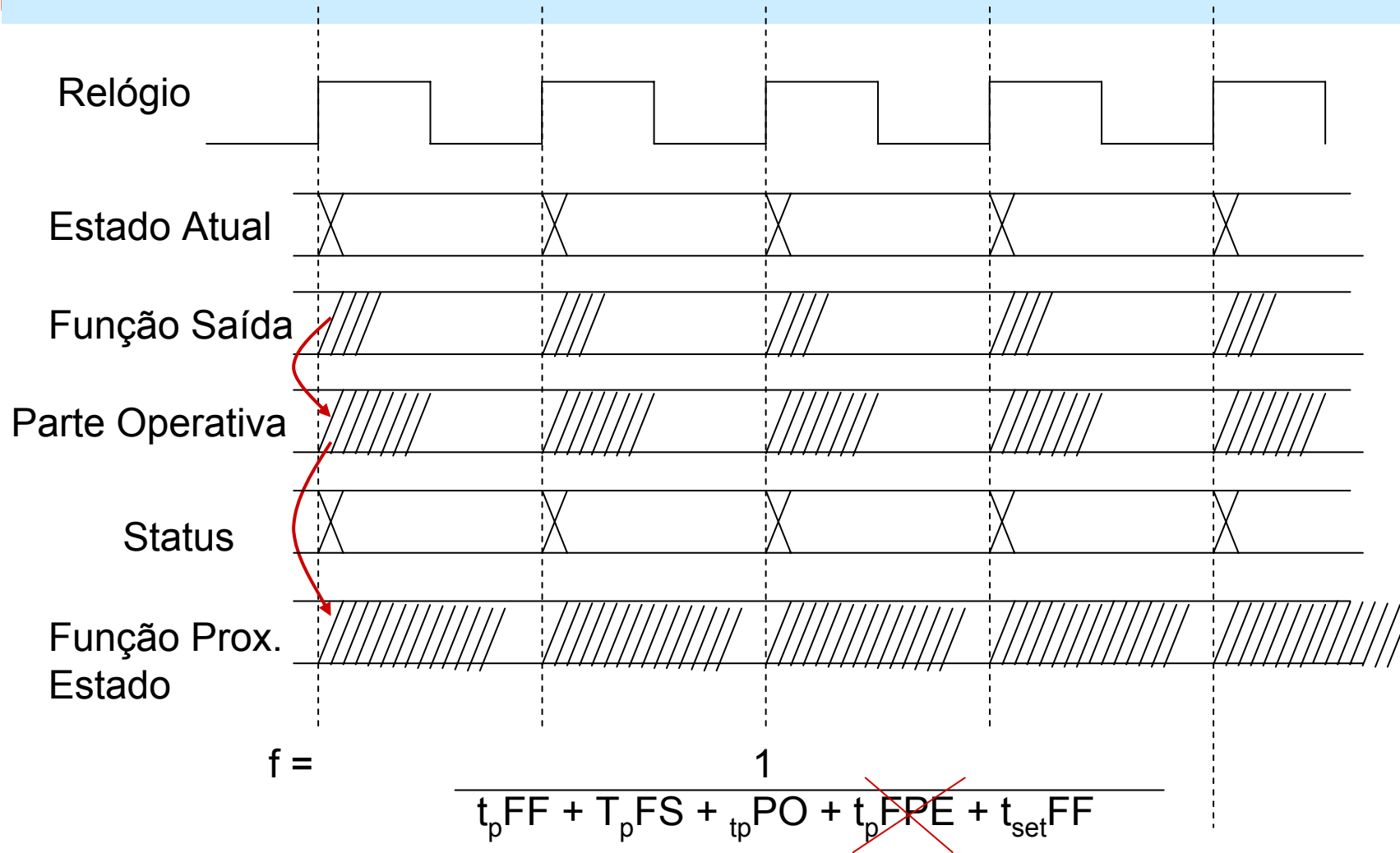


Temporização PC-PO





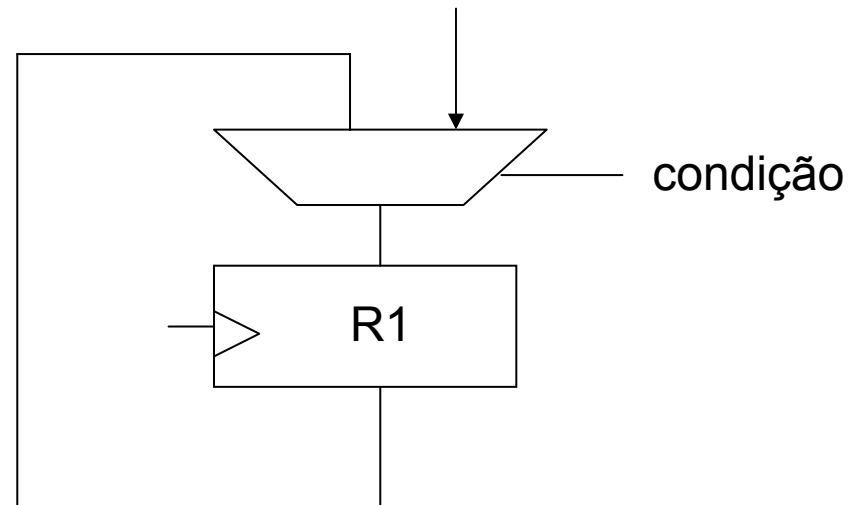
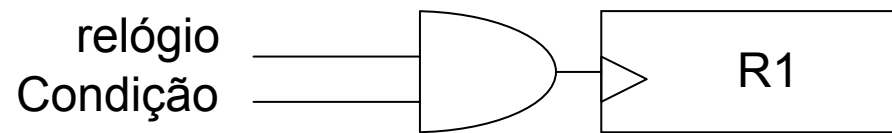
Temporização PC-PO



Maior numero de estados na PC

Controle de Carga do Registrador

Aula
2



Exemplo: Extrator de Raiz Quadrada

Aula
2

Numeros inteiros e positivos

Ler X

-- inicialização

R=1; d=2; s=4, t=1;

Enquanto (t==1) {

r=r+1;

d=d+2;

s=s+d+1;

t=ac(s,x);

}

devolve(r);

}

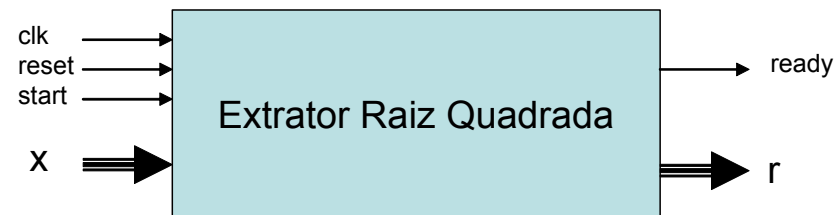
ac(a,b)

semsinal a, b;

{

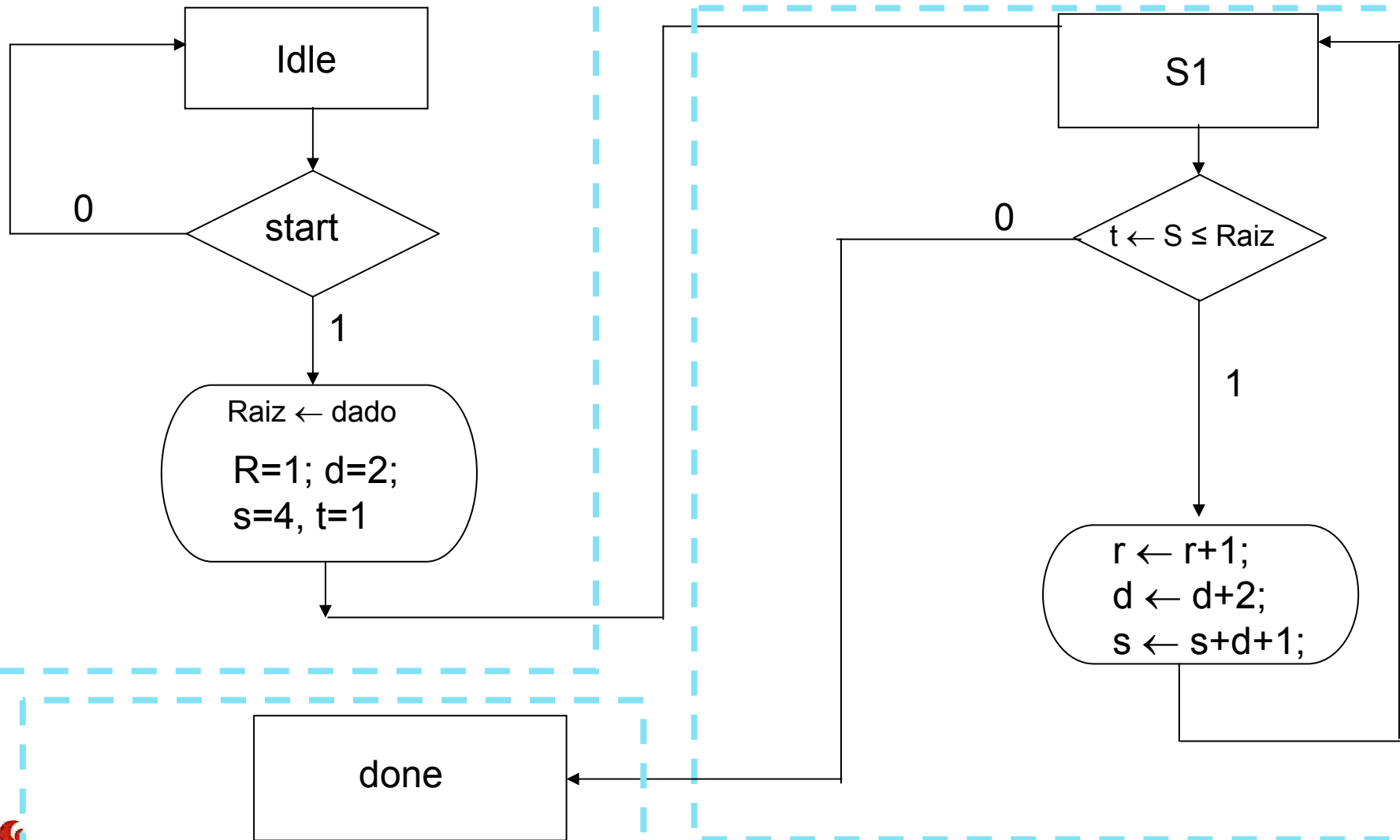
Se (a<=b) retorna (1); Senão retorna (0);

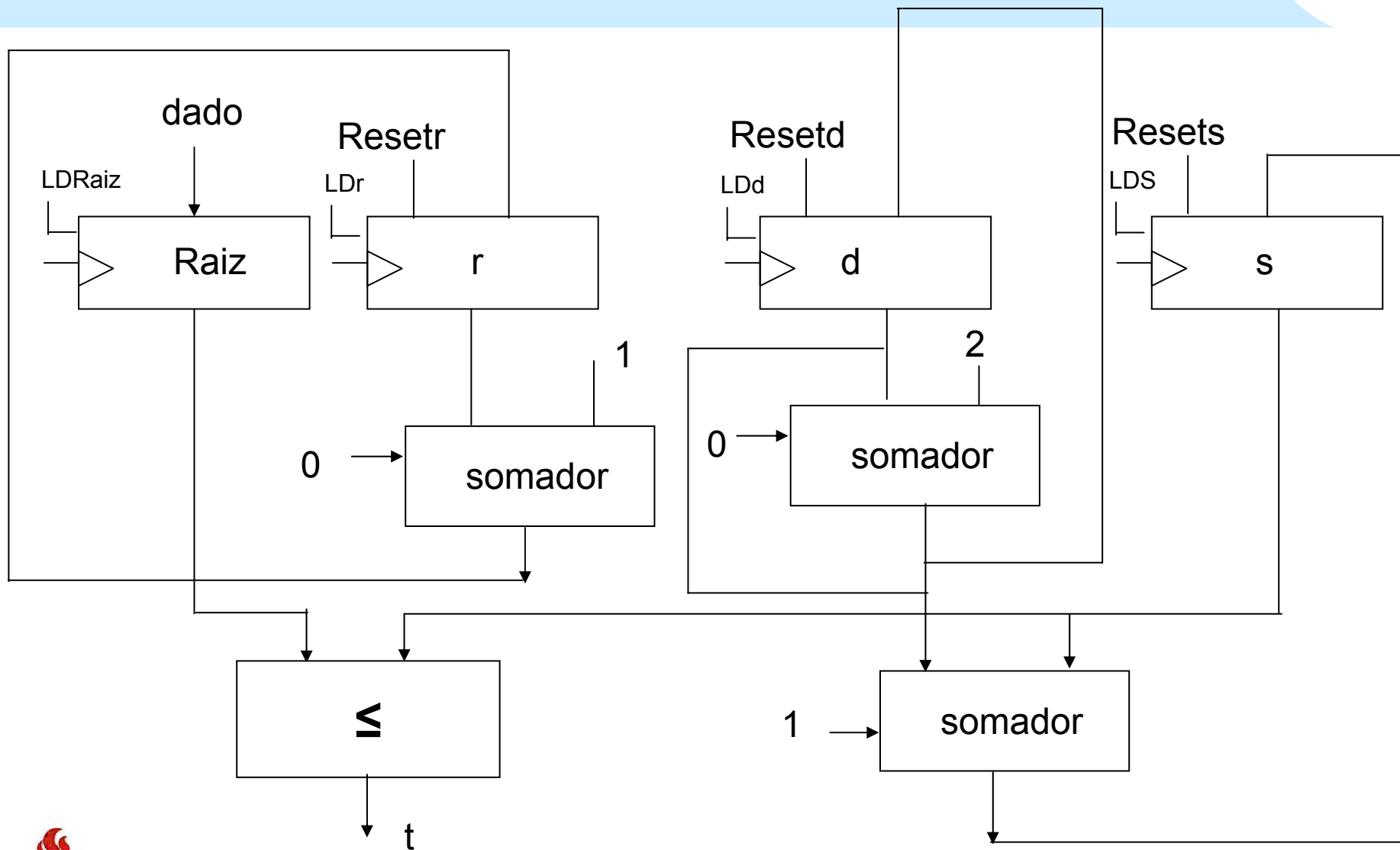
}



Exemplo: Extrator de Raiz Quadrada

Aula
2

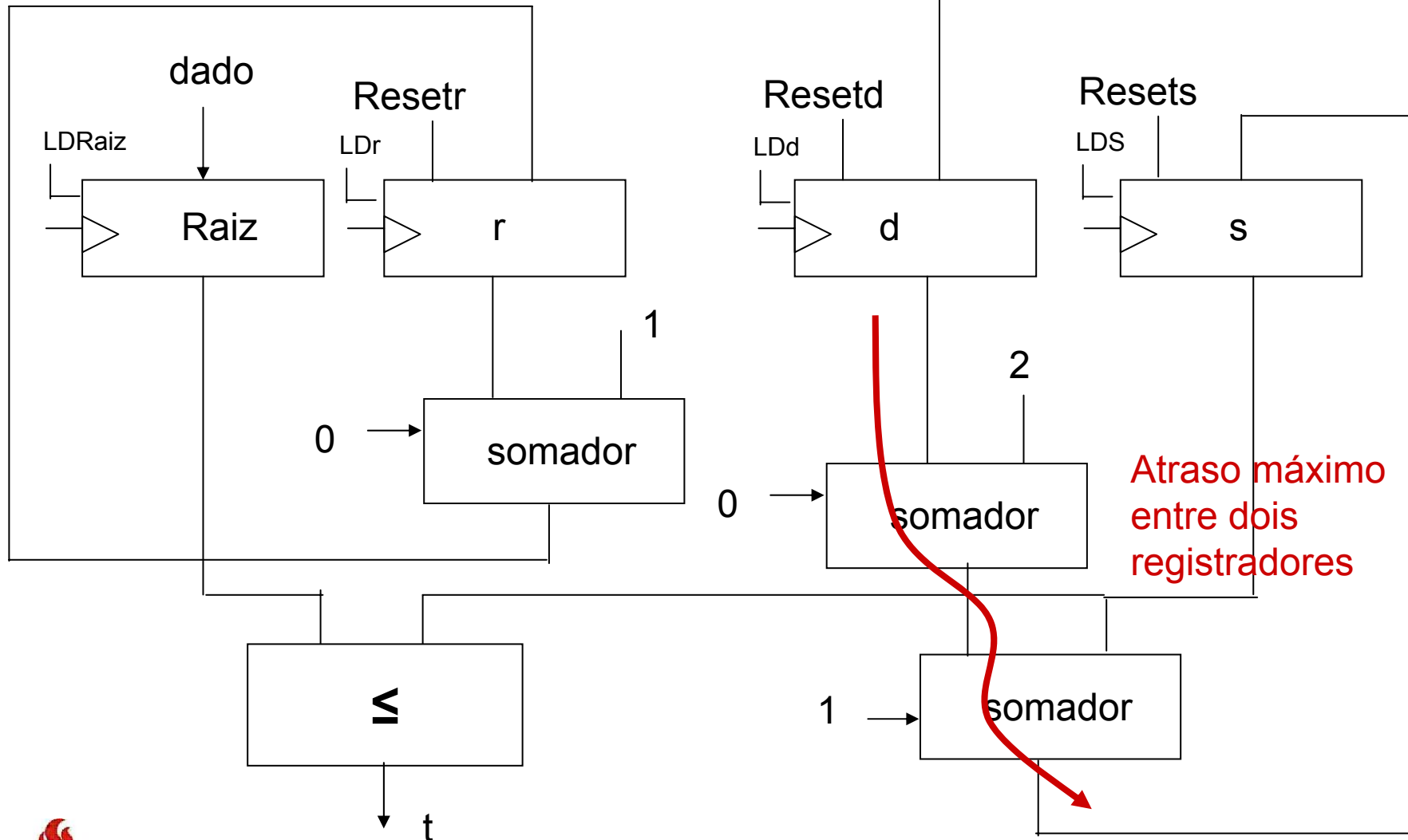


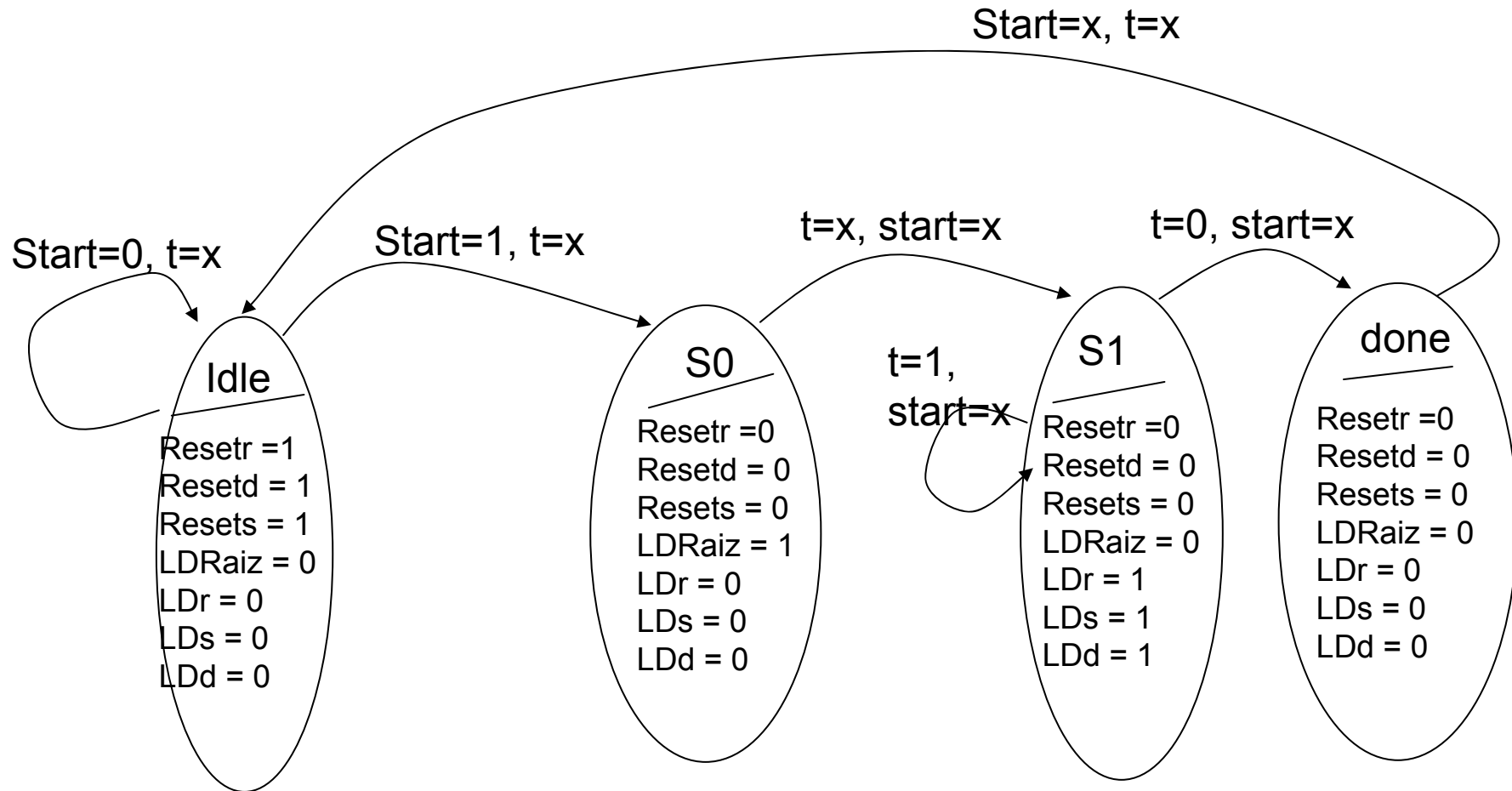


Projeto Parte Operativa

Aula
2

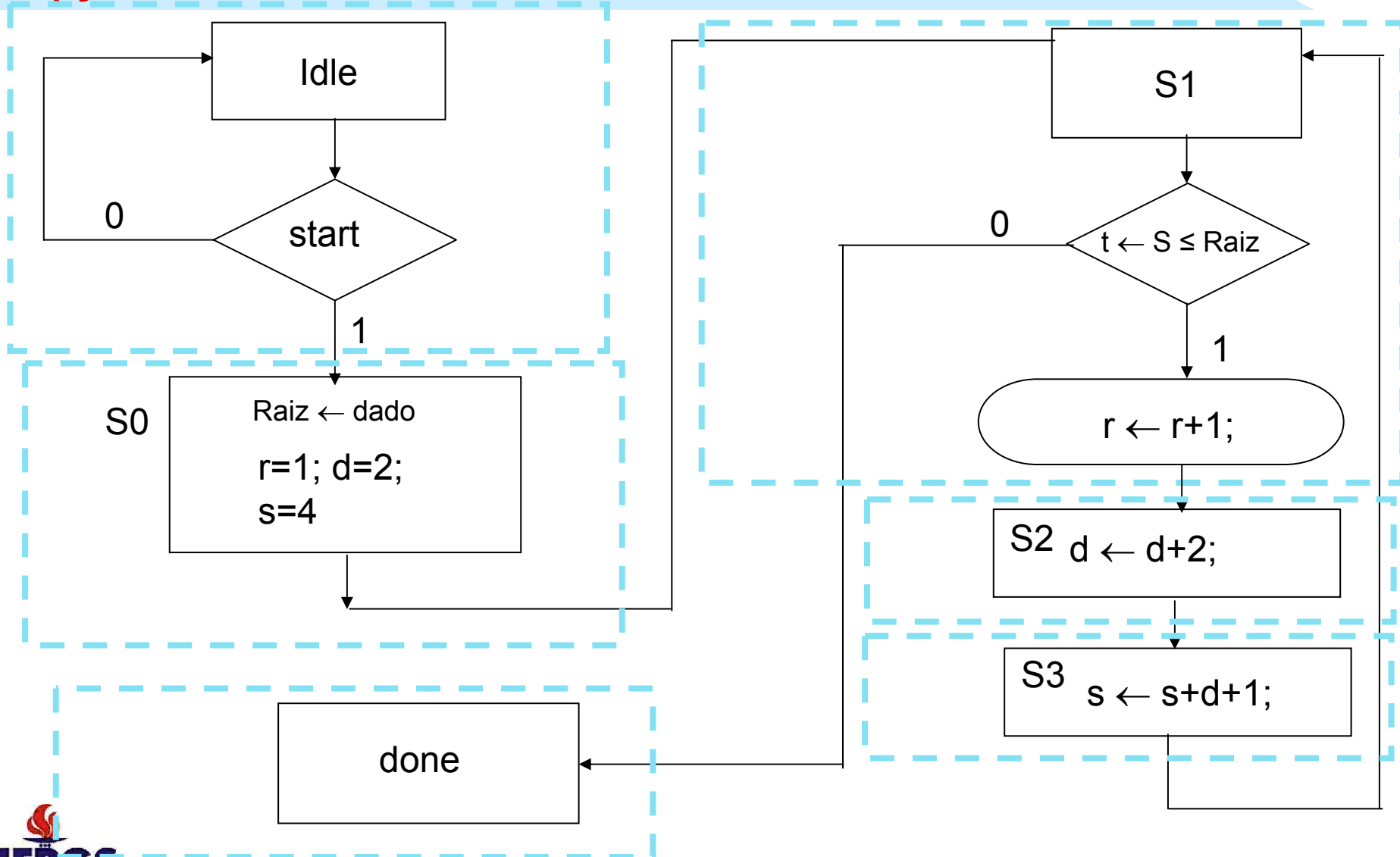
Admitindo o tempo de propagação do somador > que o comparador



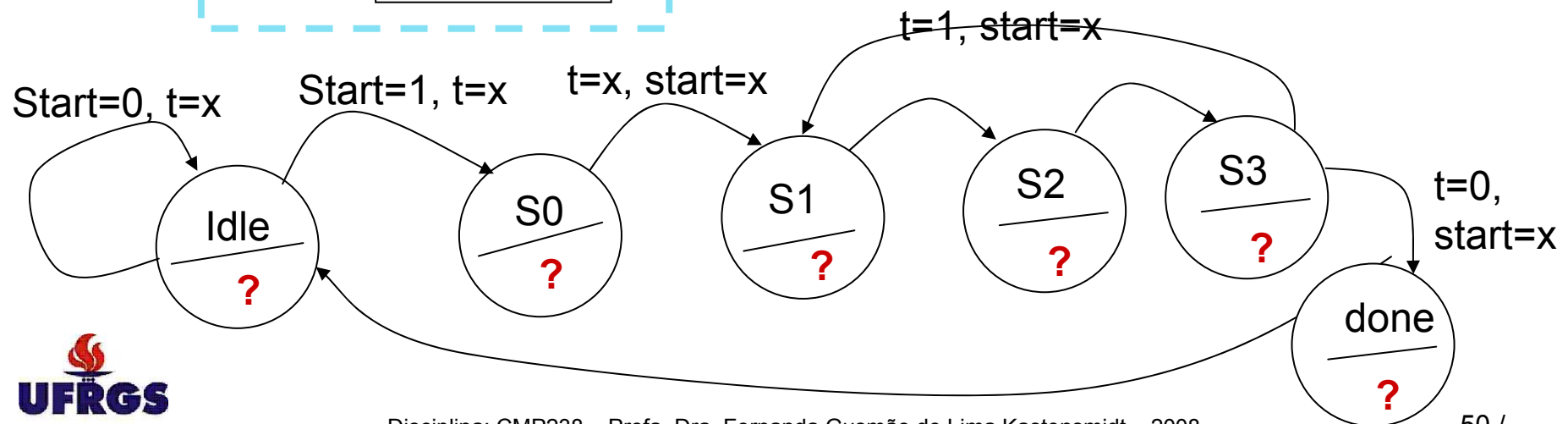
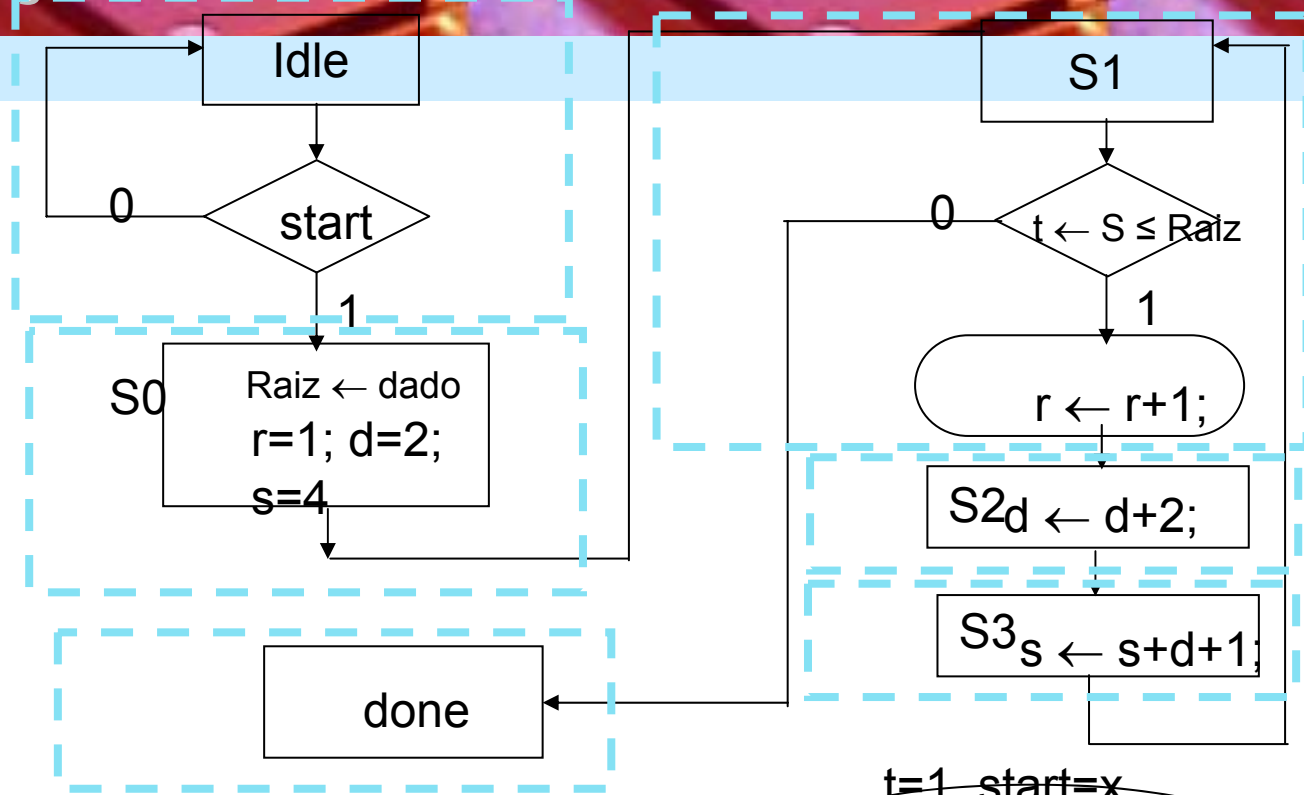


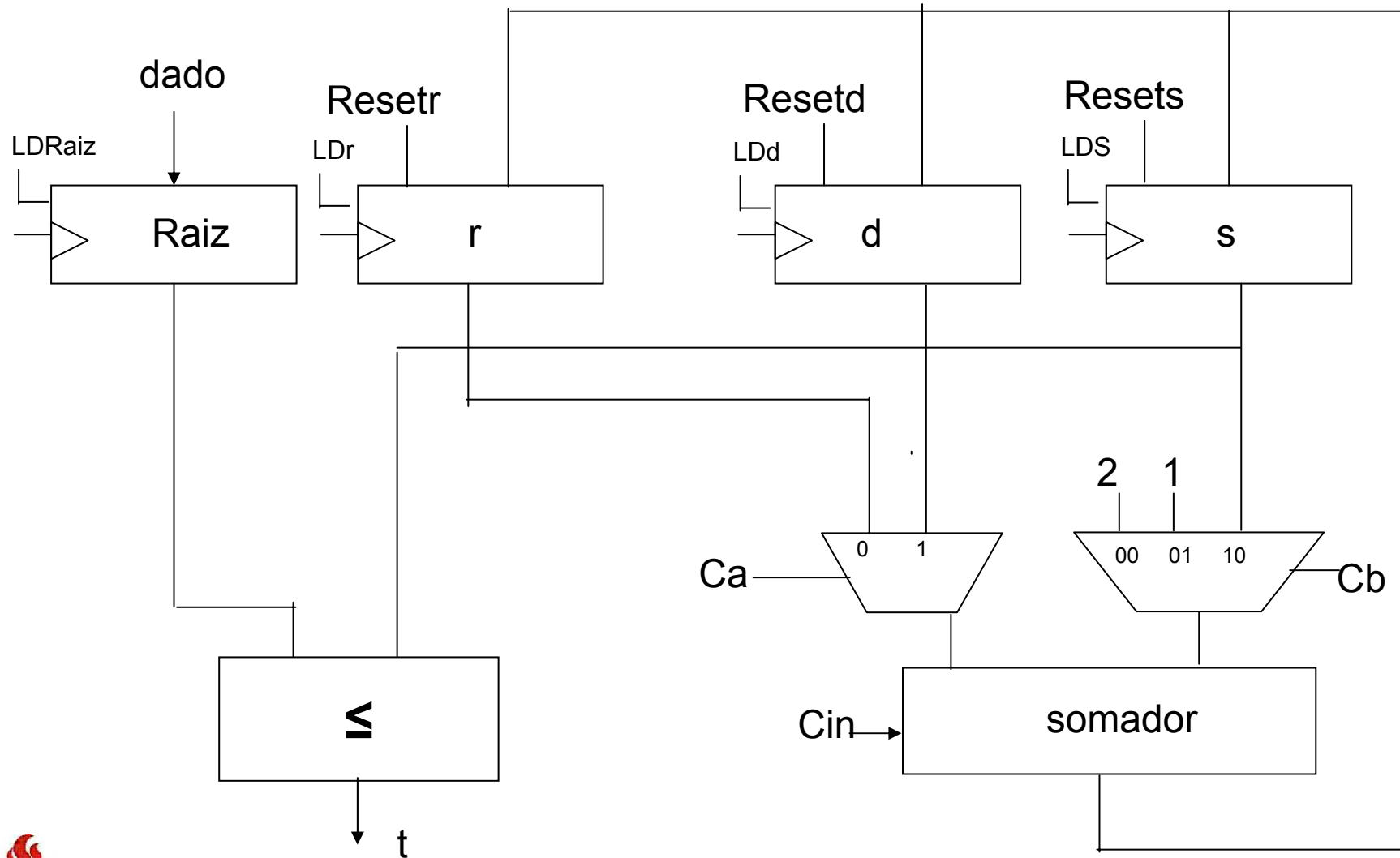
Exemplo: Extrator de Raiz Quadrada

Opção 2



Projeto Parte de Controle

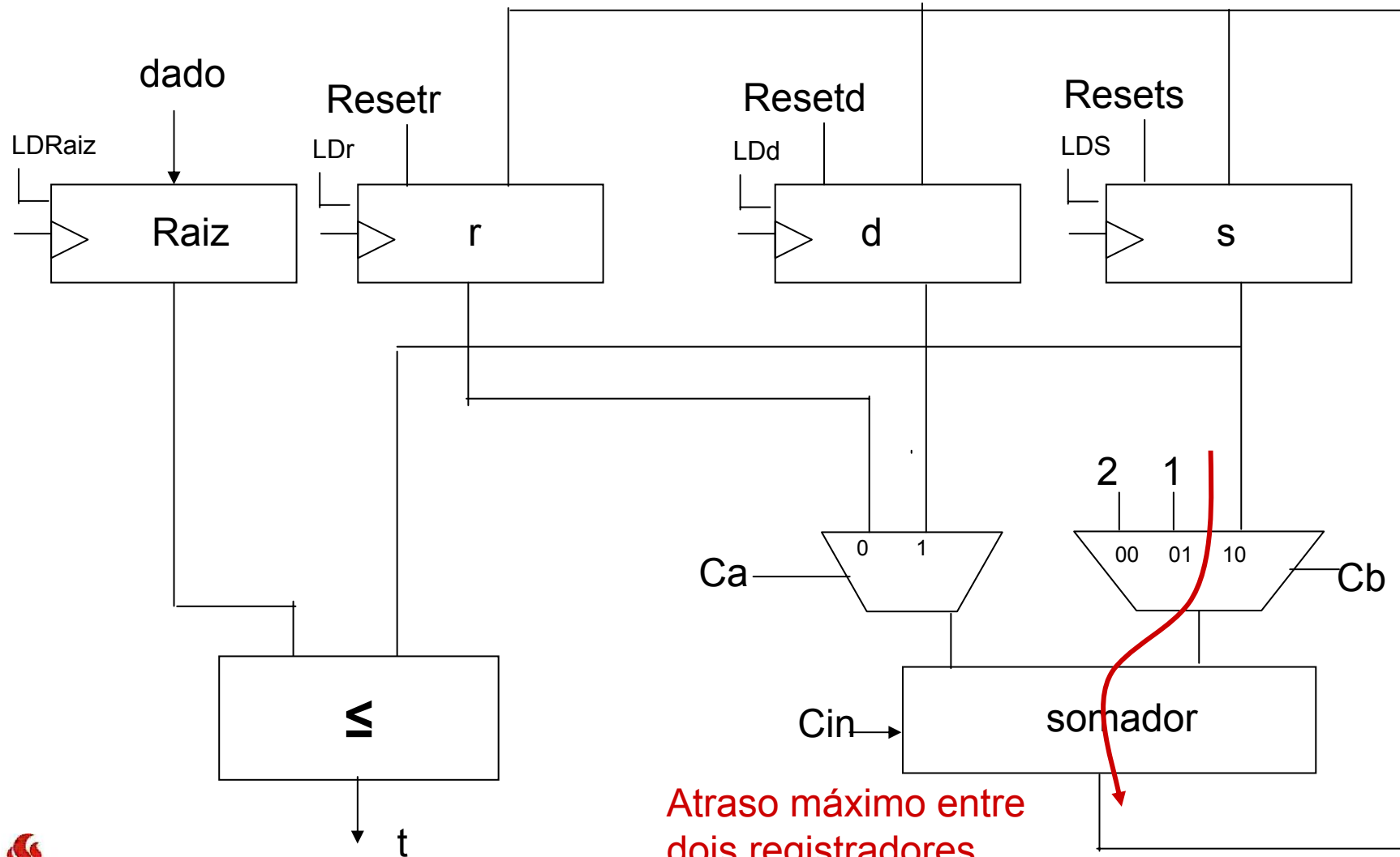




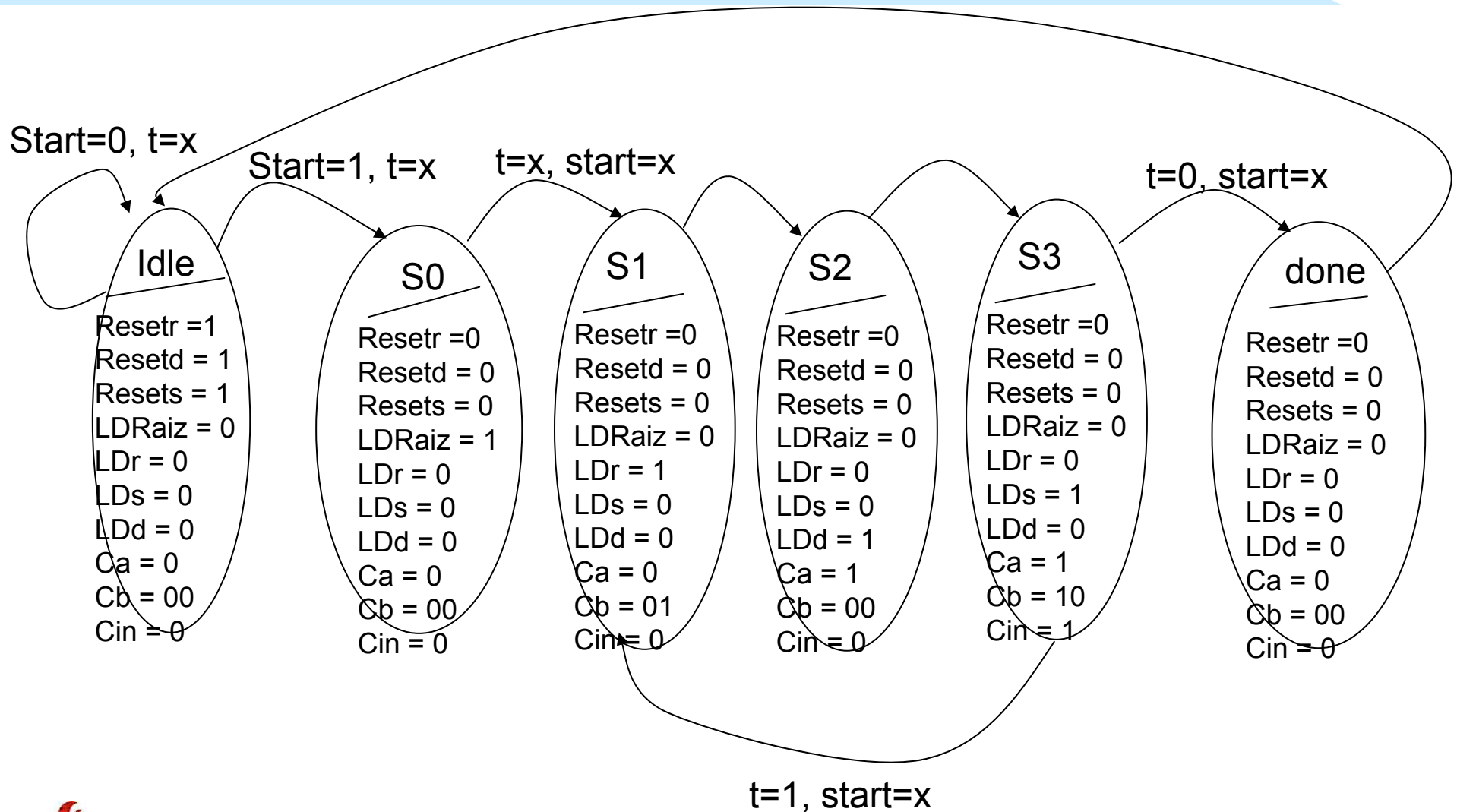
Projeto Parte Operativa

Aula
2

Admitindo o tempo de propagação do somador > que o comparador

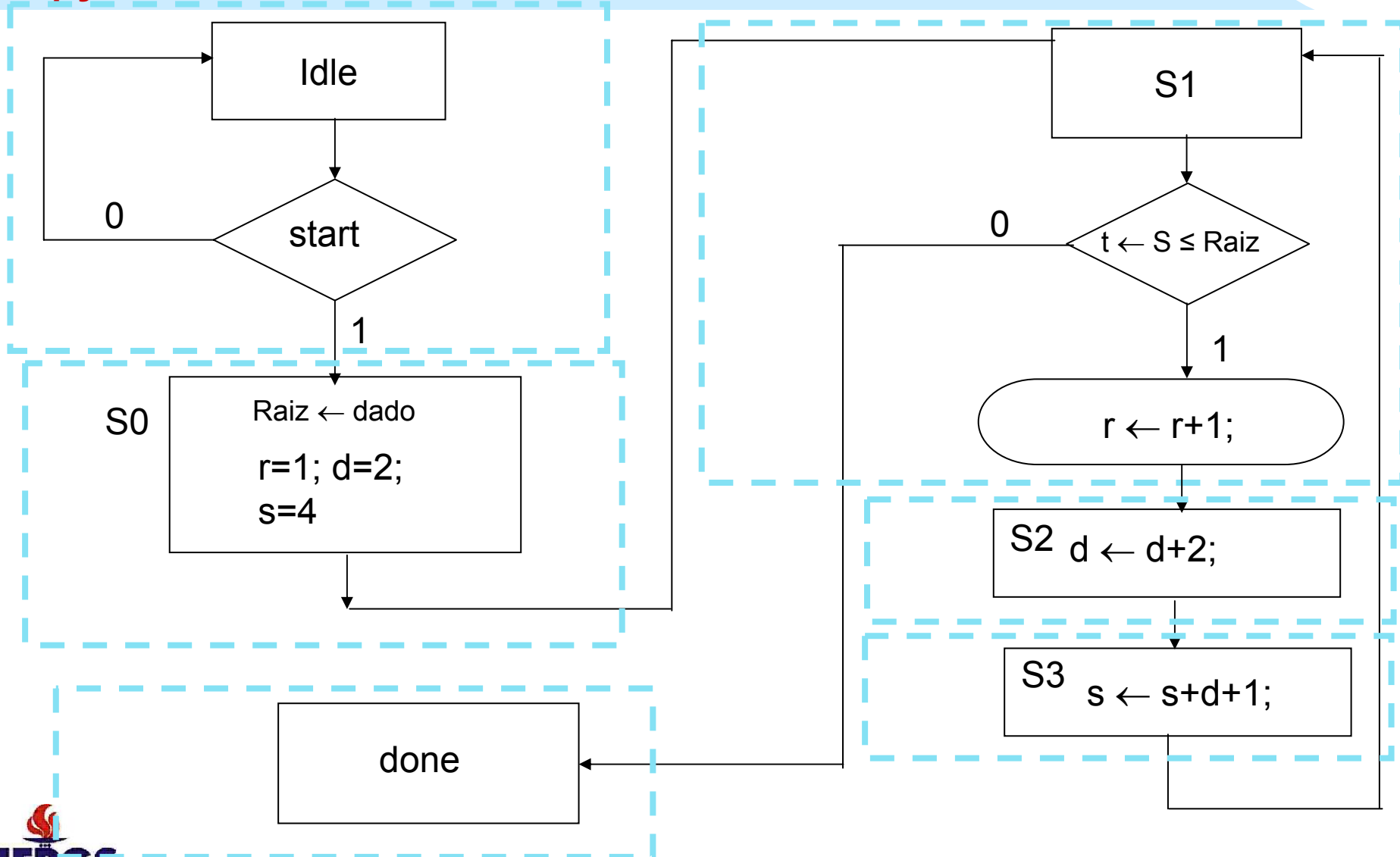


Atraso máximo entre
dois registradores

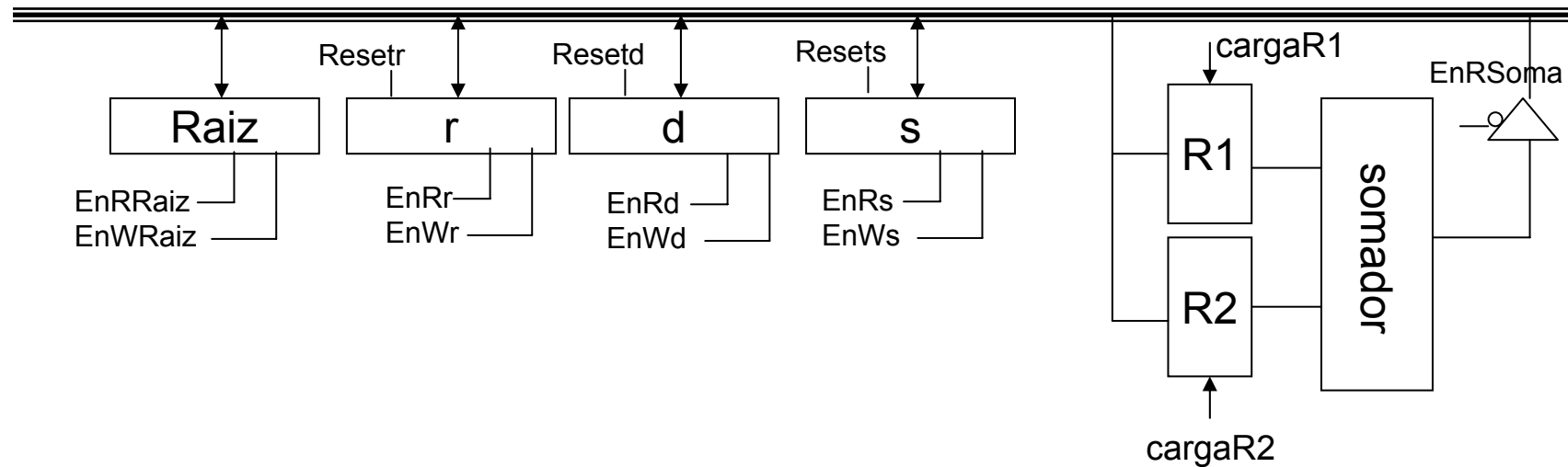


Exemplo: Extrator de Raiz Quadrada

Opção 3 - Barramento



E/S



1 transferência por ciclo