



Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Curso de Graduação em Ciências da Computação



Sistemas Digitais

INE 5406

Aula 2-T

**1. Projeto de ULA. Circuitos Digitais e Níveis de Abstração.
Funcionamento e características temporais de registradores.
Cadenciamento com sinal de relógio (clock) e análise de *timing*.**

Prof. José Luís Güntzel
guntzel@inf.ufsc.br

www.inf.ufsc.br/~guntzel/ine5406/ine5406.html

1. Projeto de Unidade Lógico-Aritmética

▶ O Mercado de Eletrônica de Consumo Dominância do Segmento *Personal Mobile Devices* (PMDs)

Computação de Propósito Geral



Computação de Propósito Específico



- Portabilidade (tamanho e peso)
- Alta capacidade de processamento (=alto desempenho)
- Baixíssimo consumo de energia

1. Projeto de Unidade Lógico-Aritmética

▶ O Mercado de Eletrônica de Consumo Dominância do Segmento *Personal Mobile Devices* (PMDs)

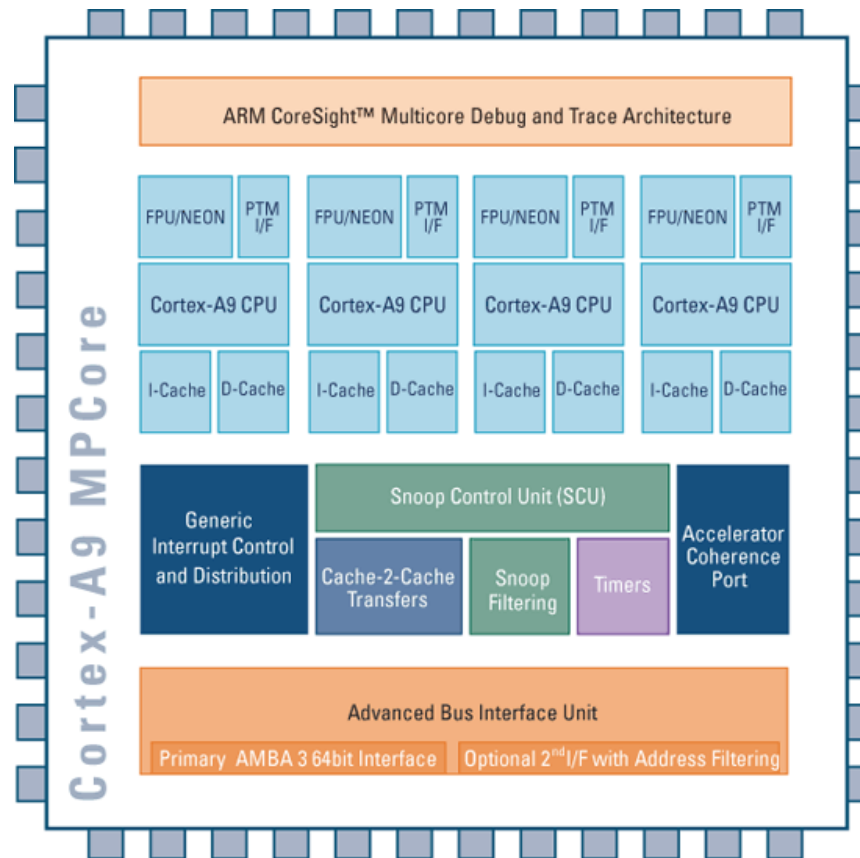


Usam MPSoCs (*Multi-Processor Systems-on-a-Chip*), fornecidos por, ou licenciados de terceiros



1. Projeto de Unidade Lógico-Aritmética

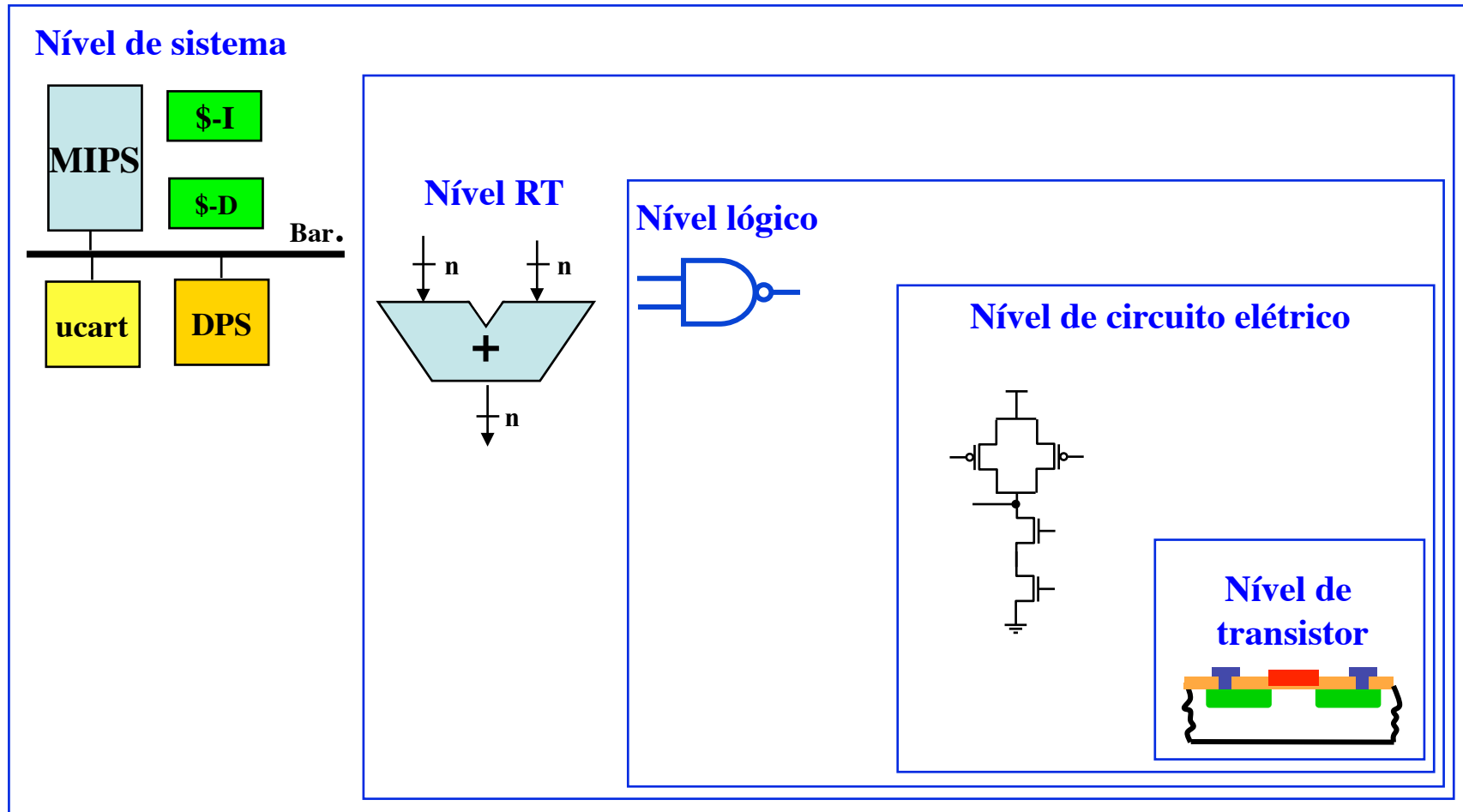
▶ O Mercado de Eletrônica de Consumo Exemplo de MPSoC: ARM Cortex™ A9



<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0407f/index.html>

1. Projeto de Unidade Lógico-Aritmética

▶ Circuitos Digitais e Níveis de Abstração

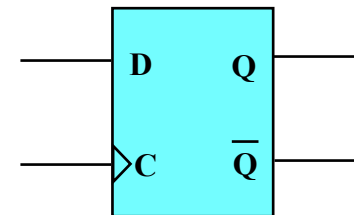
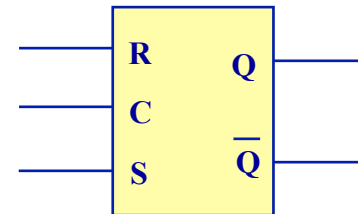
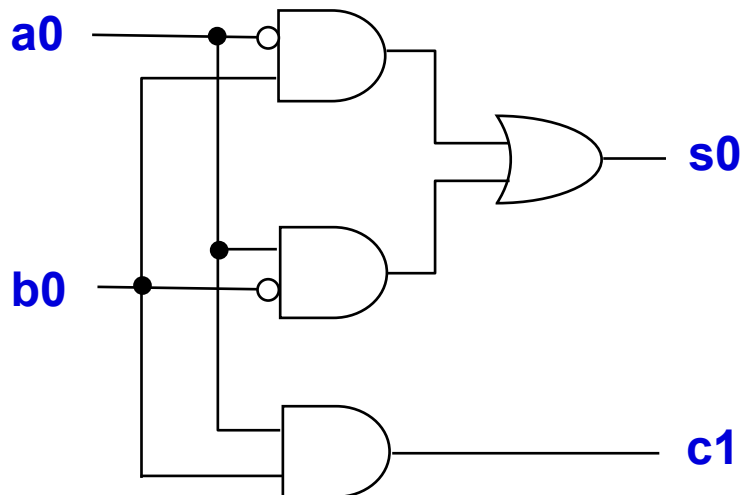


1. Projeto de Unidade Lógico-Aritmética

▶ Circuitos Digitais e Níveis de Abstração

Nível Lógico:

- Circuitos vistos como associações de portas lógicas, latches e flip-flops (conforme visto em EEL5105).
- Variáveis binárias de um bit cada.

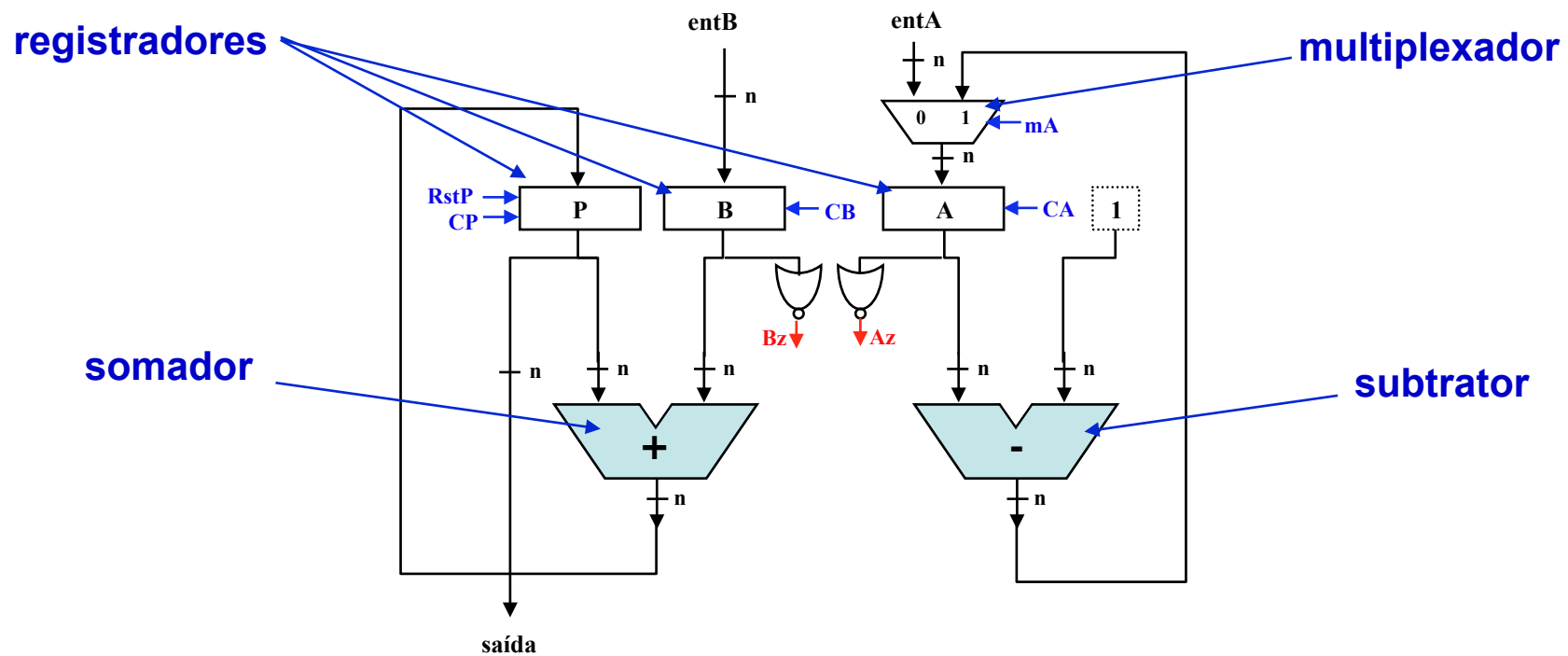


1. Projeto de Unidade Lógico-Aritmética

▶ Circuitos Digitais e Níveis de Abstração

Nível RT (Register Transfer):

- Circuitos vistos como associações de componentes, feitos de portas lógicas (objeto desta disciplina).

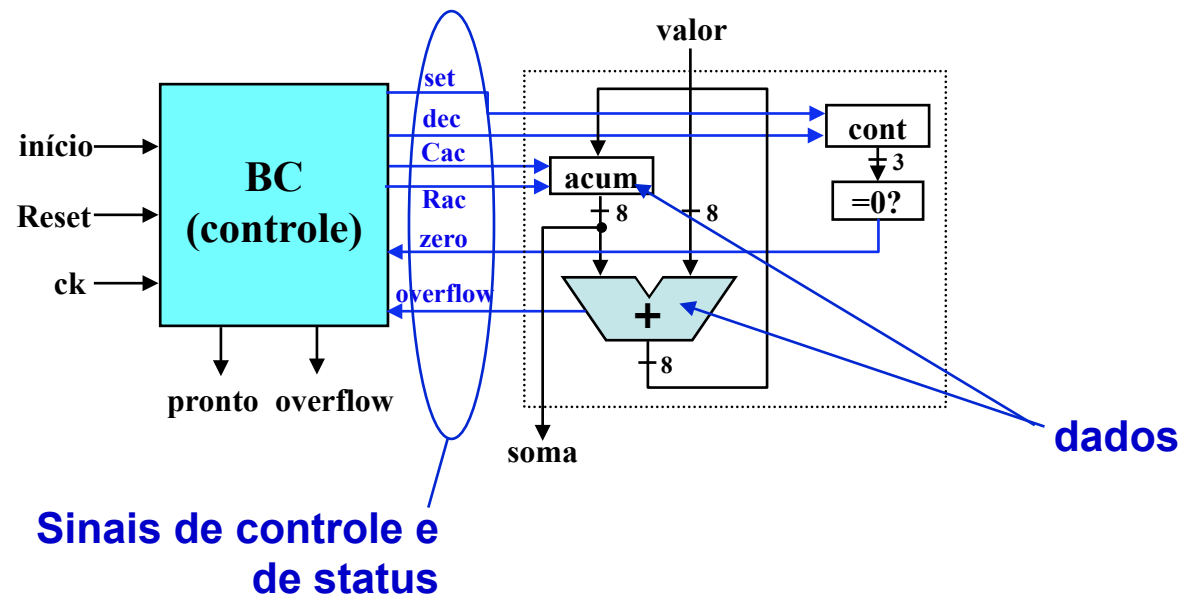


1. Projeto de Unidade Lógico-Aritmética

▶ Circuitos Digitais e Níveis de Abstração

Nível RT (Register Transfer):

- Operações lógicas e aritméticas sobre vetores de bits, os quais representam números.
- Sinais de controle e de status.



1. Projeto de Unidade Lógico-Aritmética

▶ Circuitos Digitais e Níveis de Abstração

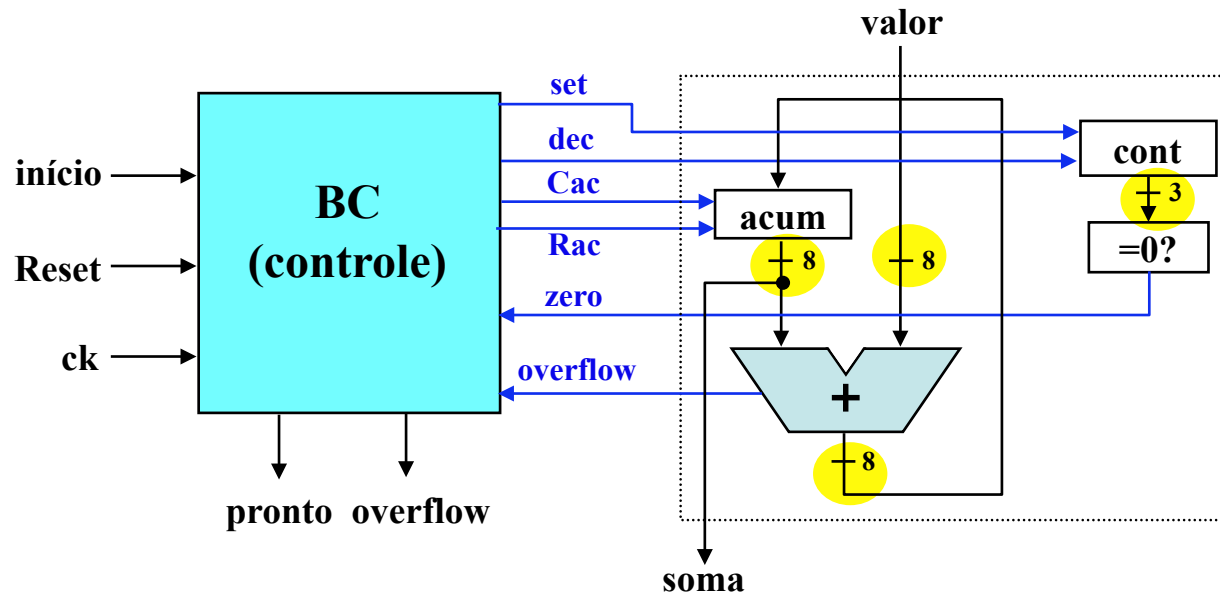
Matéria	Nível de abstração	Componentes
Circuitos Digitais	Lógico	<ul style="list-style-type: none">• Portas lógicas• Latches• Flip-flops
Sistemas Digitais	RT (register-transfer)	<ul style="list-style-type: none">• Somadores• Subtratores• ULAs (= unidades funcionais compostas)• Registradores• Memórias• Multiplexadores• Decodificadores• Bloco de controle

1. Projeto de Unidade Lógico-Aritmética

► Representando Dados em Circuitos Digitais

Exemplo: um circuito mais complexo.

- Fios que representam variáveis de mais de um bit devem ser identificados!



1. Projeto de Unidade Lógico-Aritmética

▶ Tipos de Circuitos Digitais

1. Circuitos Combinacionais:

Não são capazes de reter os sinais (dados) gerados em suas saídas (tampouco os sinais que são aplicados em suas de entradas). “As saídas dependem apenas das entradas.”

Usados para construção de:

- Circuitos aritméticos e lógicos (somadores, subtratores, deslocadores, ULAs, comparadores, multiplexadores, decodificadores ...)**
- Lógica de próximo estado e lógica de saída, nos circuitos de controle**

1. Projeto de Unidade Lógico-Aritmética

▶ Tipos de Circuitos Digitais

2. Circuitos Sequenciais:

São capazes de armazenar os sinais de entrada (enquanto estiverem alimentados com energia). “As saídas dependem das entradas e do estado que está armazenado.”

Usados para construção de:

- Registradores (latches e flip-flops)

1. Projeto de Unidade Lógico-Aritmética

▶ Flip-flop D Disparado Pela Borda Ascendente (ou sensível à borda ascendente ou à borda de subida...)

símbolo

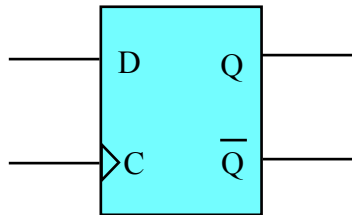
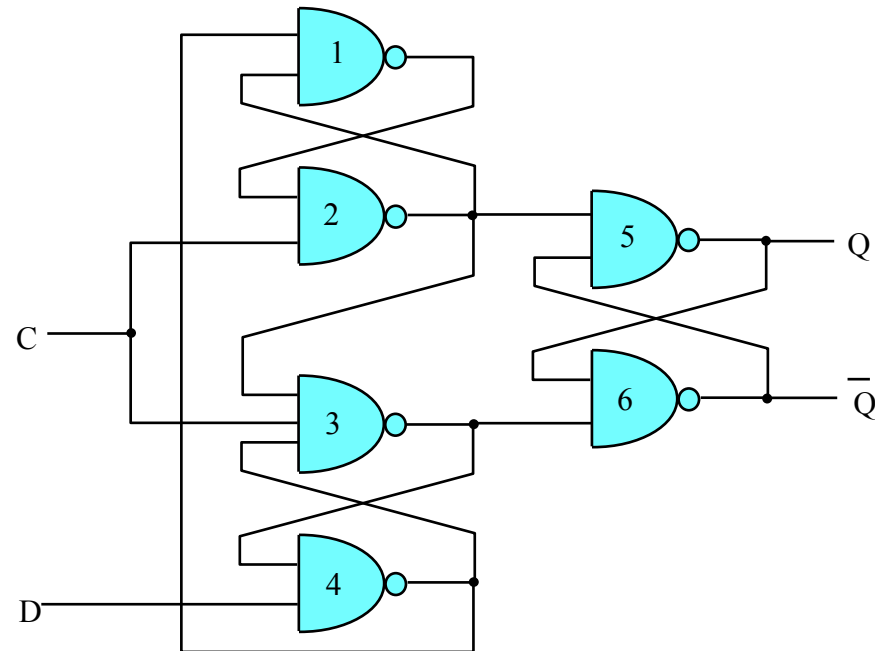


tabela de transição de estados

C	D	Q_{t+1}
$\neq \uparrow$	X	Q_t
\uparrow	0	0
\uparrow	1	1

circuito com portas nand



1. Projeto de Unidade Lógico-Aritmética

▶ Flip-flop D Disparado Pela Borda Ascendente Exemplo de funcionamento

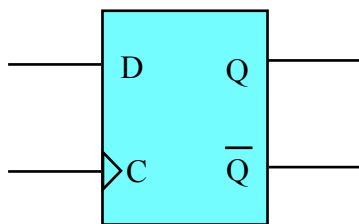
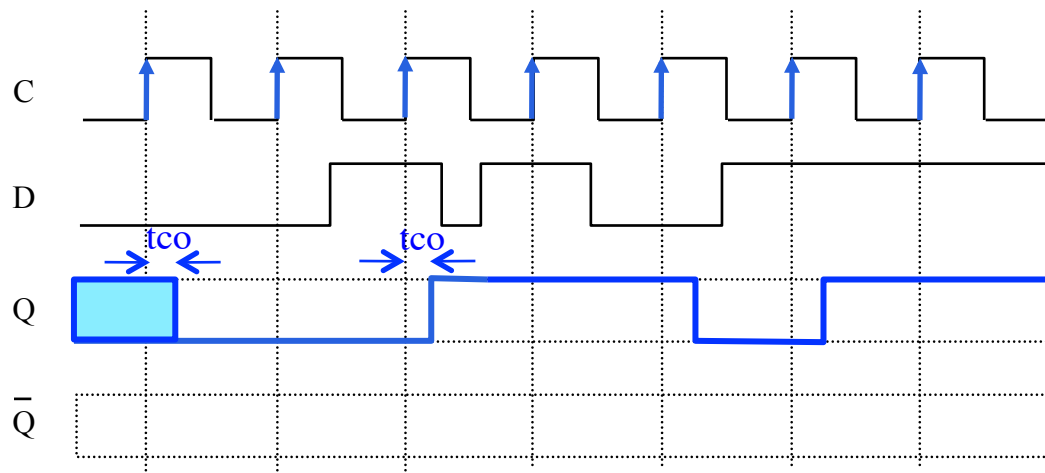


tabela de transição
de estados

C	D	Q_{t+1}
$\neq \uparrow$	X	Q_t
\uparrow	0	0
\uparrow	1	1



Obs: Nesta disciplina assumiremos que o atraso da saída Q' é idêntico ao atraso da saída Q . Assim, a saída Q' será o “espelho” da saída Q .

1. Projeto de Unidade Lógico-Aritmética

▶ Flip-flop D Disparado Pela Borda Descendente Exemplo de funcionamento

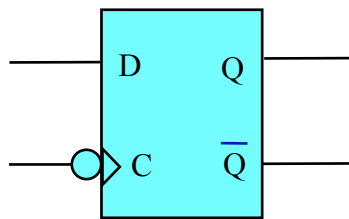
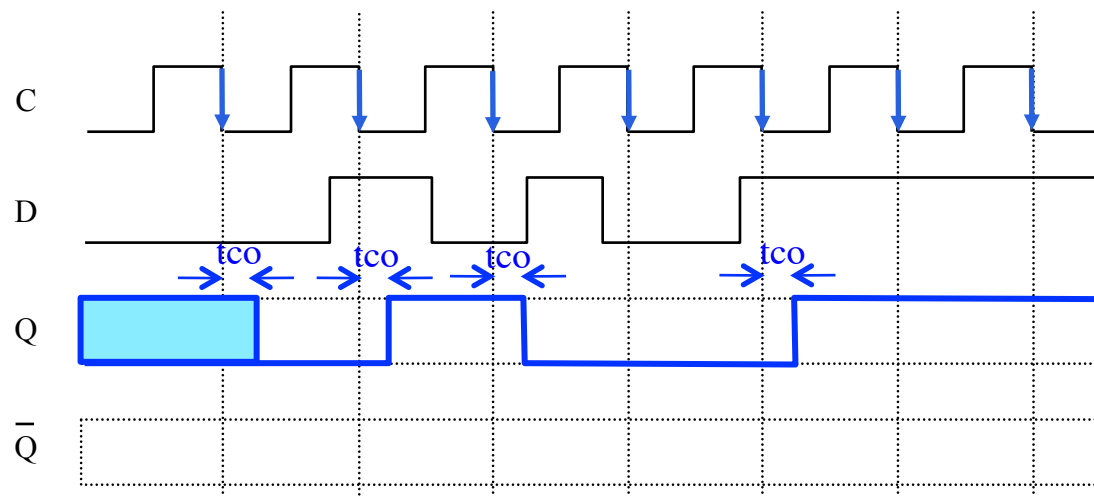


tabela de transição
de estados

C	D	Q_{t+1}
$\neq \downarrow$	X	Q_t
\downarrow	0	0
\downarrow	1	1



1. Projeto de Unidade Lógico-Aritmética

► Flip-flop D Disparado Pela Borda Ascendente, com Reset Assíncrono

Exemplo de funcionamento

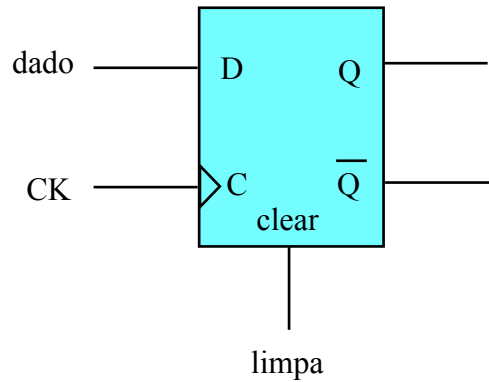
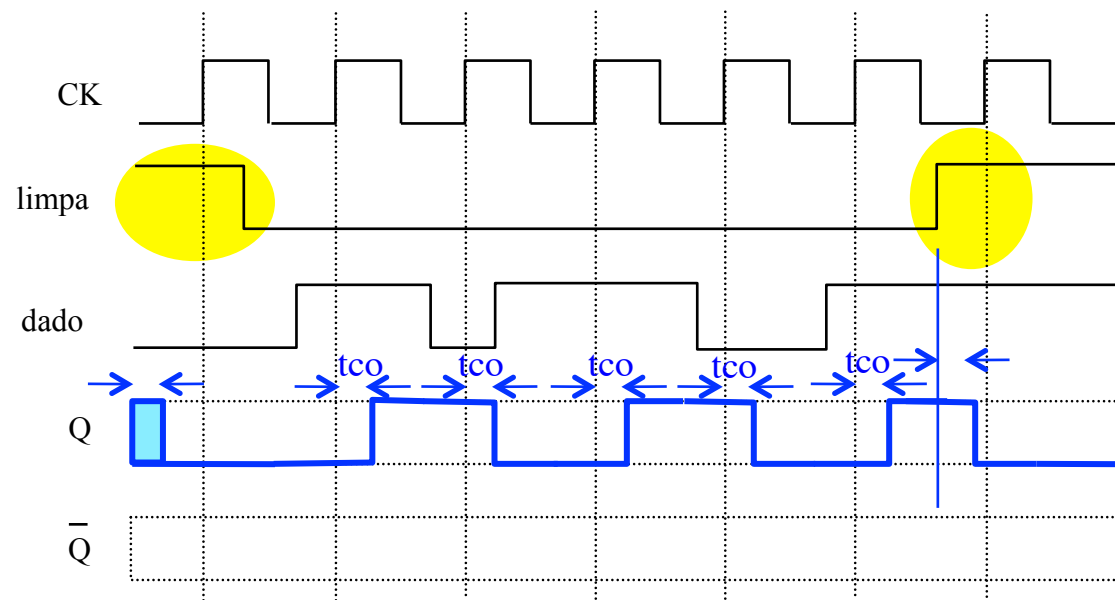


tabela de transição de estados

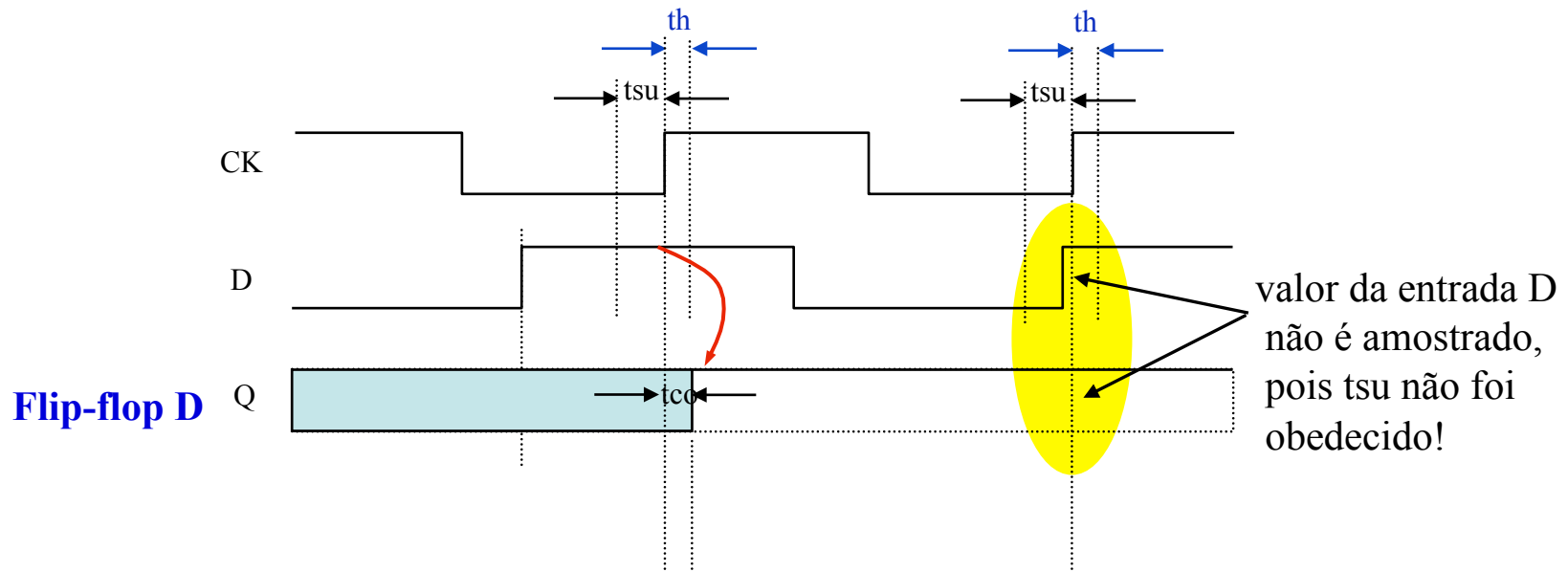
clear	C	D	Q_{t+1}
0	$\neq \uparrow$	X	Q_t
0	\uparrow	0	0
0	\uparrow	1	1
1	X	X	0



1. Projeto de Unidade Lógico-Aritmética

▶ Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida



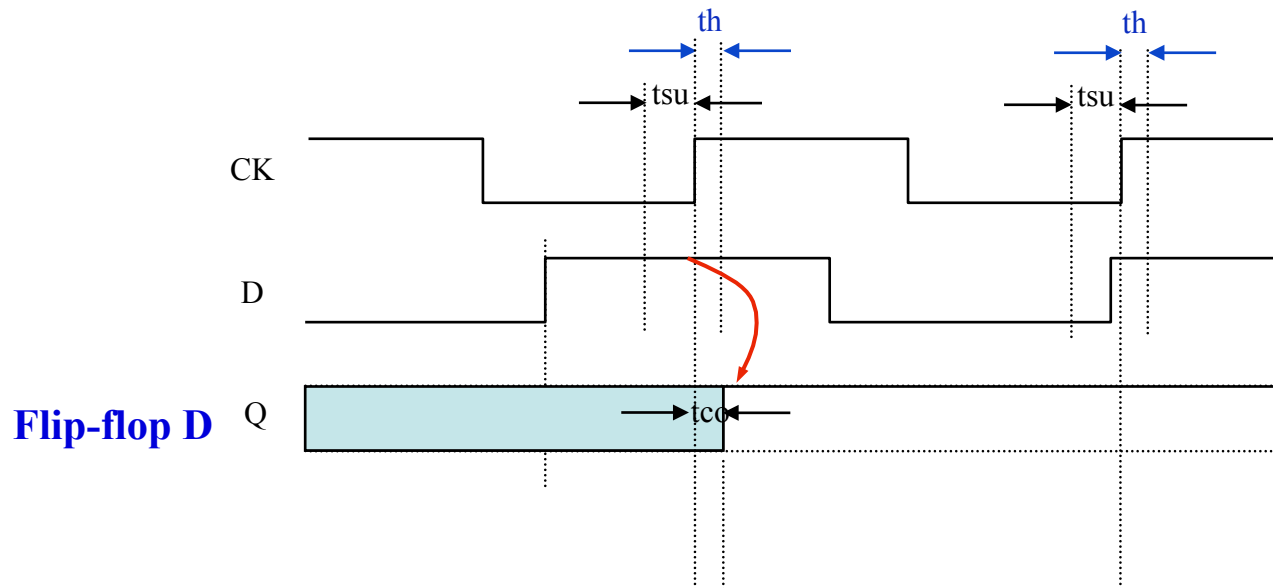
tsu = Tempo de Preparação (*setup time*)

“Tempo antes da borda ativa de ck (subida, neste caso) durante o qual a entrada D já deve estar em seu valor estável.”

1. Projeto de Unidade Lógico-Aritmética

▶ Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida



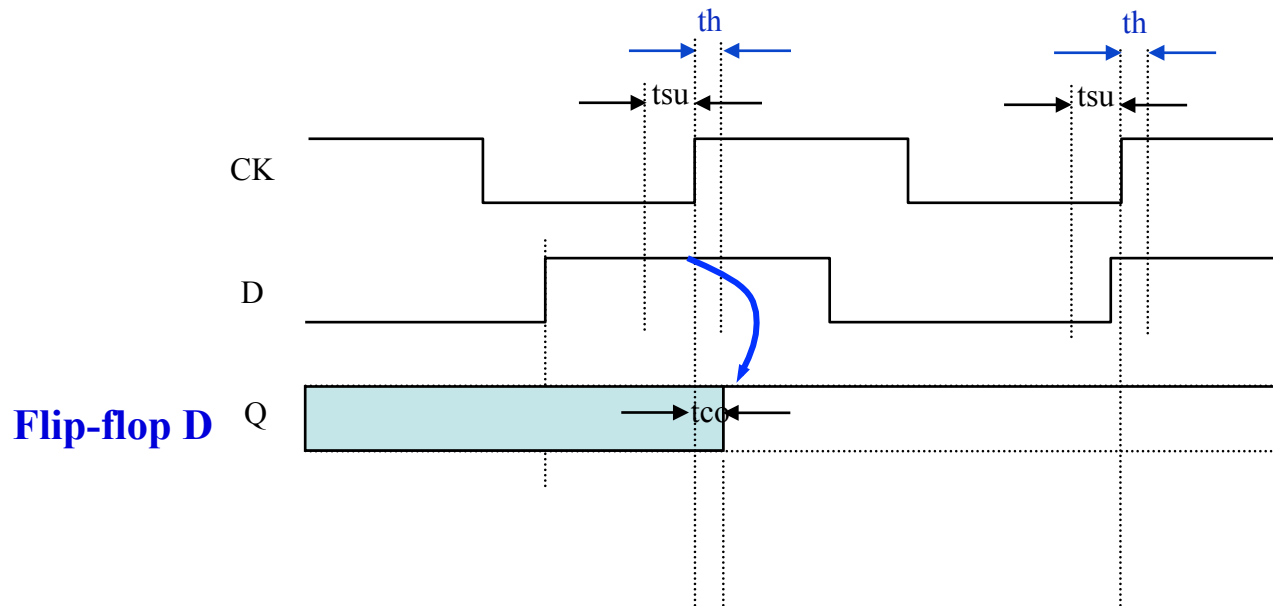
th = Tempo de Manutenção (*hold time*)

“Tempo, a partir da borda ativa de ck (subida, neste caso), durante o qual a entrada D deve permanecer estável.”

1. Projeto de Unidade Lógico-Aritmética

▶ Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida

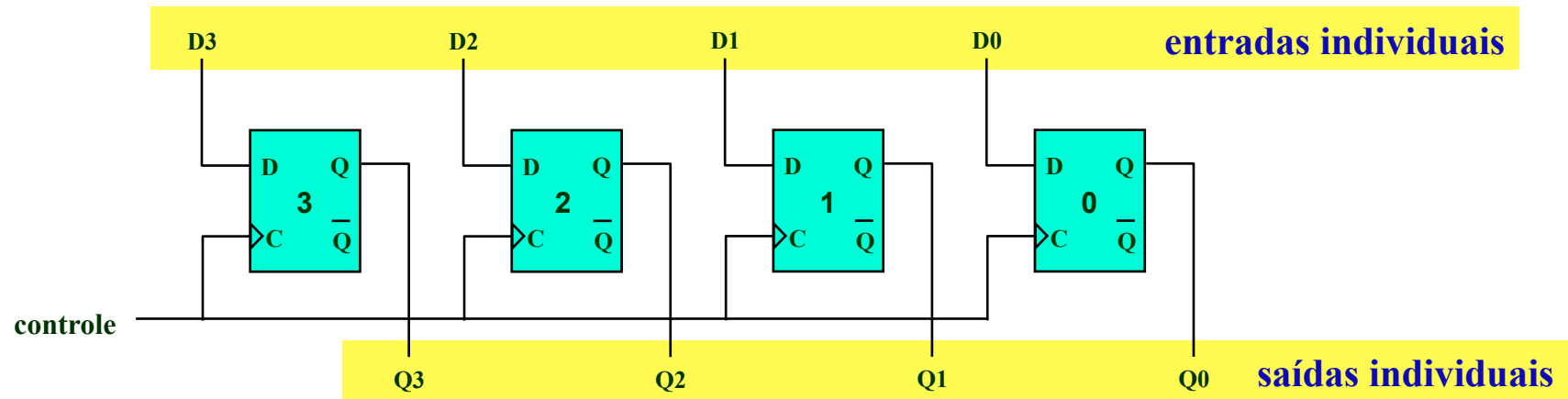


tco (ou td) = *Time from clock to output* (ou Tempo ou Atraso de Carga)

“Atraso, em relação à borda ativa de ck (subida, neste caso), para o valor amostrado a partir da entrada D aparecer nas saídas Q e Q.”

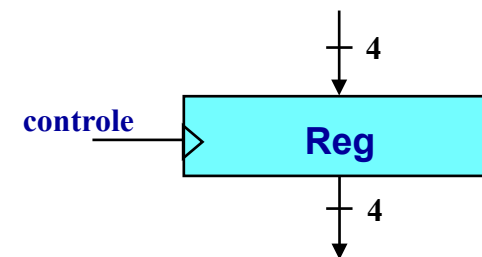
1. Projeto de Unidade Lógico-Aritmética

▶ Registrador com Carga Paralela



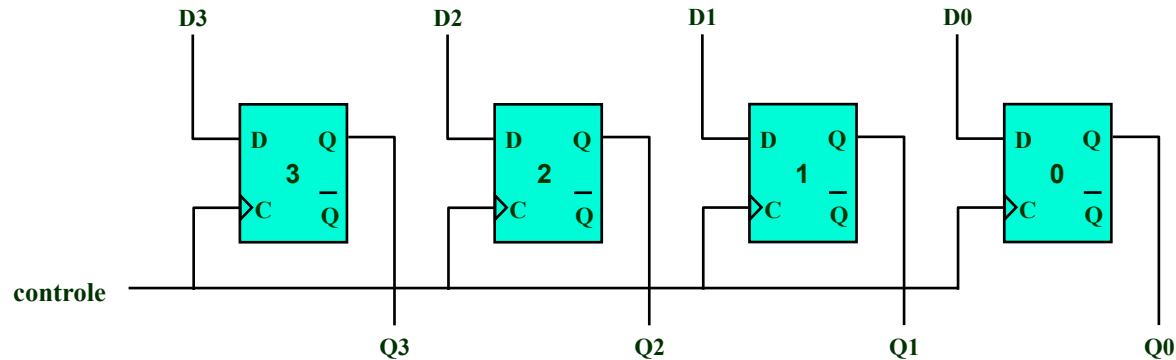
- Existe um FF para cada bit a ser armazenado
- Todos os FFs “obedecem” a um mesmo sinal de controle
- O termo “carga paralela” refere-se ao fato de existir uma entrada para cada bit, de modo que é possível carregar simultaneamente todos os bits do dado

Símbolo no nível RT

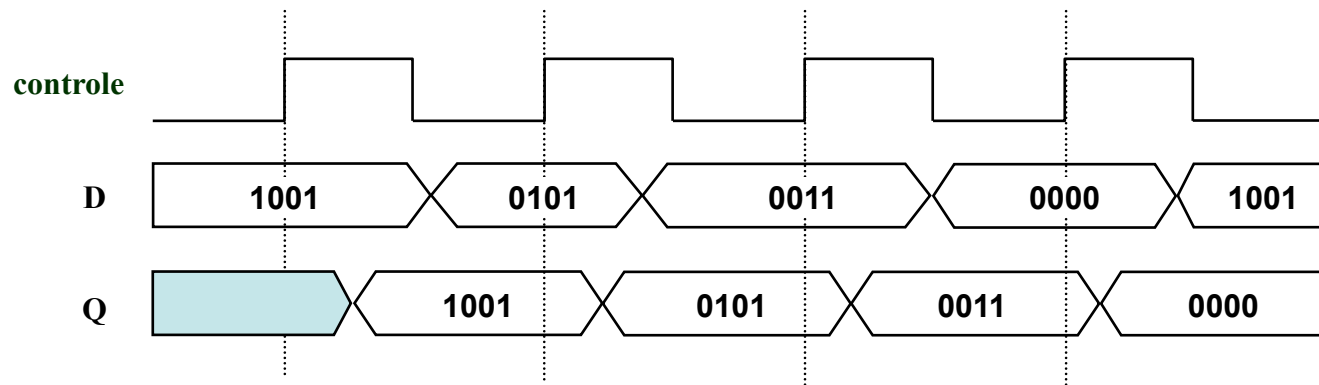


1. Projeto de Unidade Lógico-Aritmética

▶ Registrador com Carga Paralela



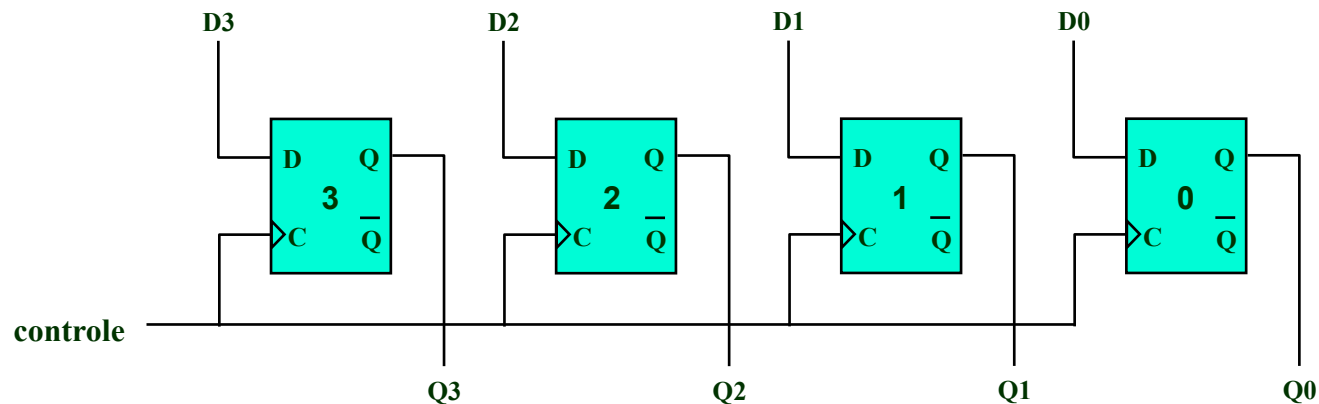
Exemplo de funcionamento (assumindo que t_{su} , t_h e t_d são satisfeitos)



A cada borda de subida de “controle” um novo dado é amostrado e fica armazenado no registrador (até a próxima borda de subida de “controle”)

1. Projeto de Unidade Lógico-Aritmética

▶ Registrador com Carga Paralela



- Peculiaridade: a cada borda de subida de “controle” um novo dado é amostrado, **mesmo que não se queira amostrar um dado novo...**
- Porém, às vezes pode ser necessário que o registrador obedeça a um sinal de controle sincronizado com o sinal de relógio. (Solução na próxima transparência...)

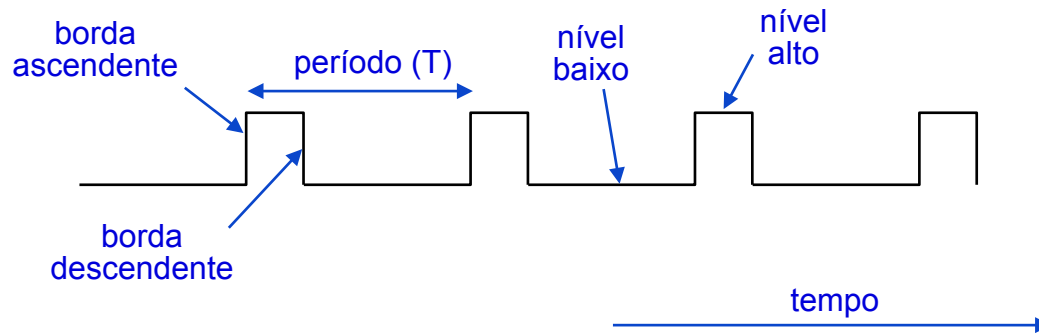
1. Projeto de Unidade Lógico-Aritmética

▶ Cadenciamento de Sistemas Digitais

- A maior parte dos sistemas digitais são sincronizados por um sinal monótono denominado **relógio** (ou *clock*, em inglês).
- Sistemas digitais cadenciados por sinal de relógio são denominados **síncronos**.
- No projeto de sistemas digitais síncronos, registradores são utilizados.

1. Projeto de Unidade Lógico-Aritmética

► Sinal de Relógio (ou *Clock*)



Nomenclatura

borda ascendente =
borda de subida =
borda positiva =
flanco de subida etc

borda descendente =
borda de descida =
borda negativa =
flanco de descida etc

Características:

- **Período (T):** tempo entre duas bordas consecutivas de mesmo tipo.
Medido em submúltiplos do segundo (ms, μ s, ns)
- **Frequência:** $f = 1/T$, medida em múltiplos do hertz (kHz, MHz, GHz)
- **Duty cycle:** $T1/T \times 100 \%$, onde T1 é o tempo entre uma borda ascendente e a borda descendente que a segue.

1. Projeto de Unidade Lógico-Aritmética

▶ Estimativa do Período do Relógio

Exemplo 1: Estime o período do relógio para um circuito combinacional cujo atraso crítico é 10 ns.

Preliminares:

$$1 \text{ ns (1 nanossegundo)} = 1 \times 10^{-9} \text{ s}$$

$$T = 1/f \Rightarrow f = 1/T$$

$$1/1\text{s} = 1 \text{ Hz}$$

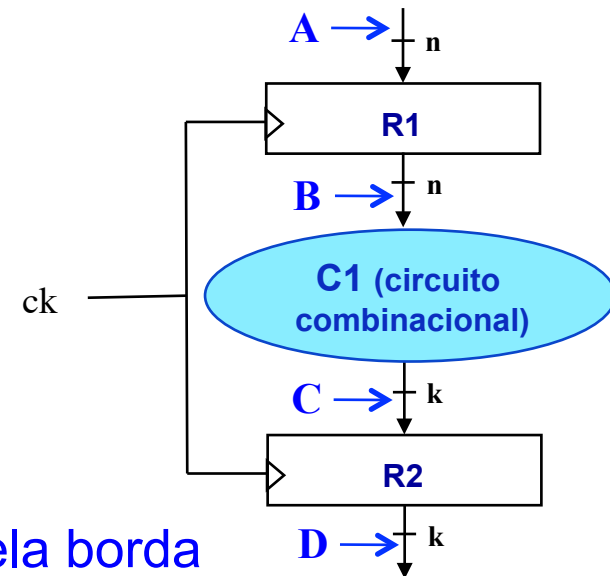
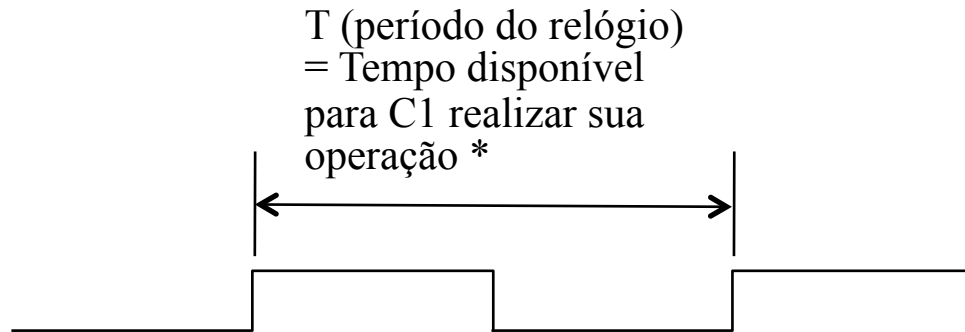
Cálculo:

$$f = 1/ (10 \times 10^{-9}) \text{ Hz} = 0,1 \times 10^9 \text{ Hz} = 100 \times 10^6 \text{ Hz} = \\ = \mathbf{100 \text{ MHz}}$$

1. Projeto de Unidade Lógico-Aritmética

▶ Cadenciamento de Sistemas Digitais

Registadores são usados para criar “barreiras temporais” que isolam os circuitos combinacionais



* (obs) :

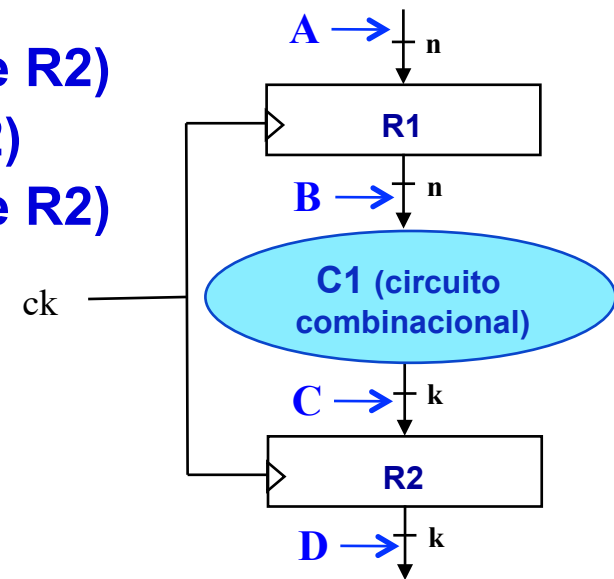
1. Supondo que R1 e R2 sejam disparados pela borda de subida de ck.
2. Aproximação grotesca; falta considerar as características temporais dos registradores R1 e R2.

1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $t_{su_{R1}} = t_{su_{R2}} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $t_{h_{R1}} = t_{h_{R2}} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $t_{co_{R1}} = t_{co_{R2}} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $t_{d_{c1}} = 2\text{ns}$ (atraso crítico (máximo) de C1)

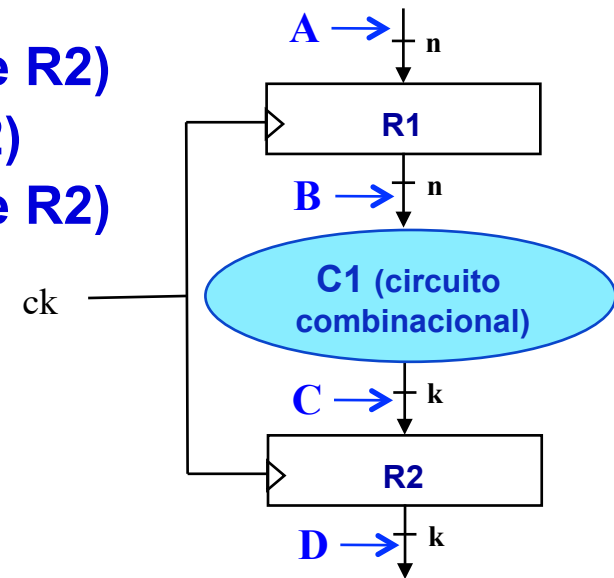
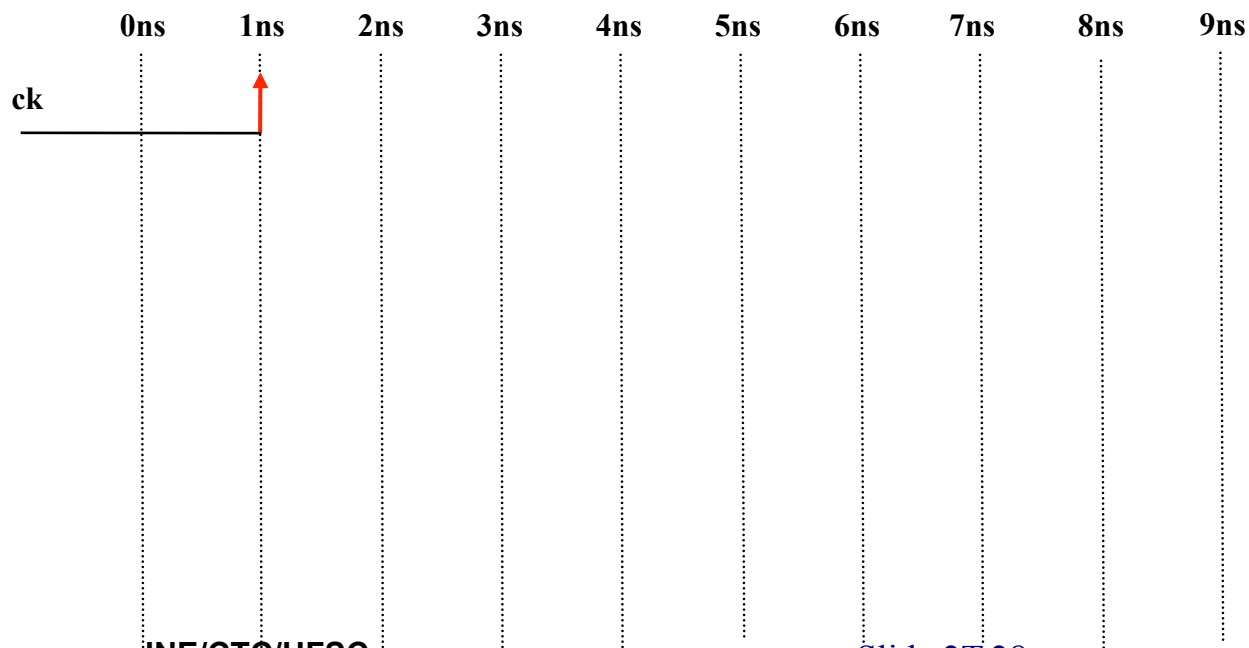


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)

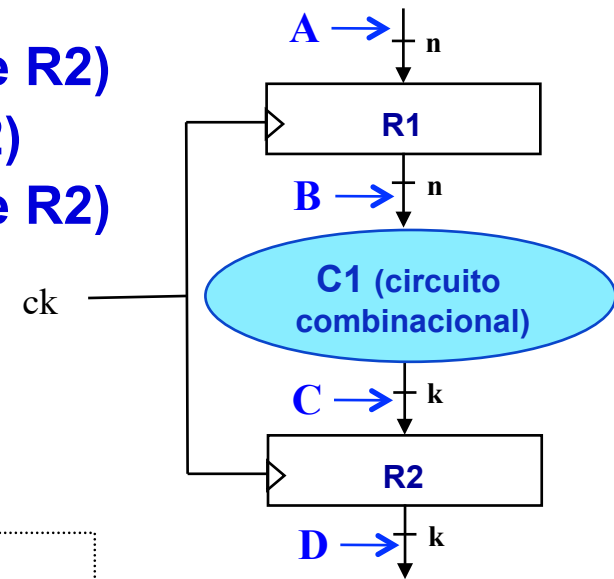
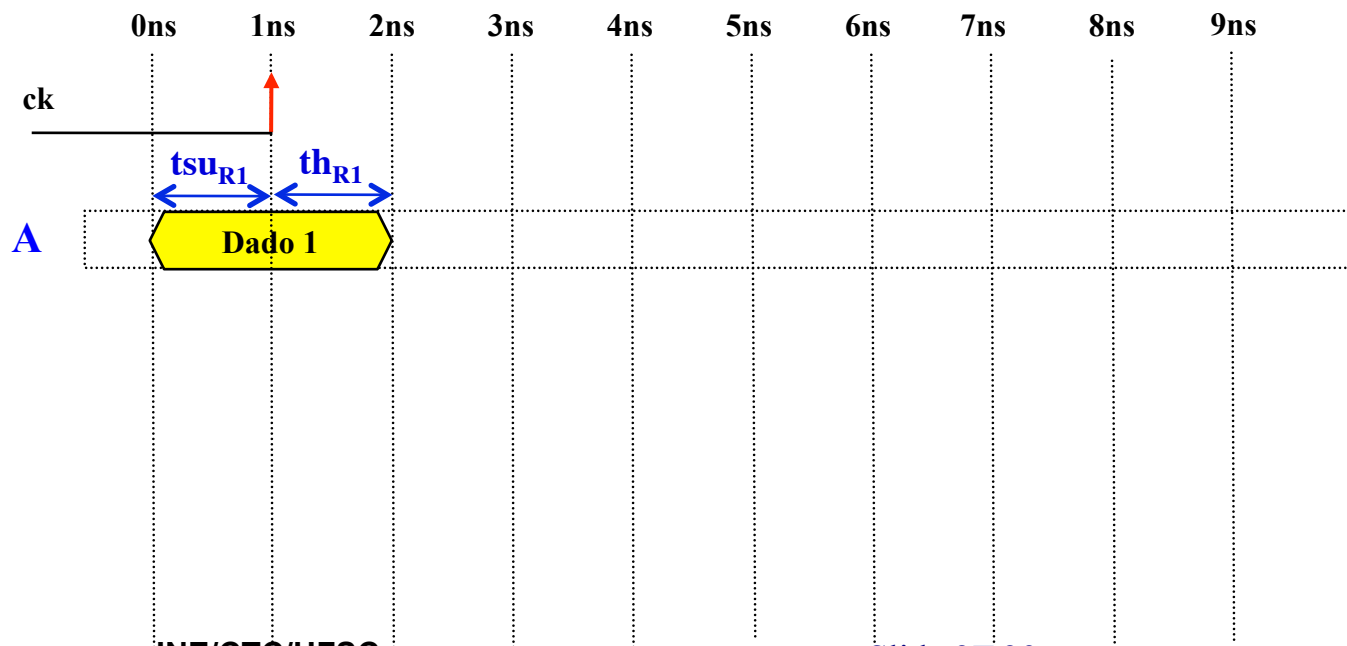


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $t_{su_{R1}} = t_{su_{R2}} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $t_{h_{R1}} = t_{h_{R2}} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $t_{co_{R1}} = t_{co_{R2}} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $t_{d_{C1}} = 2\text{ns}$ (atraso crítico (máximo) de C1)

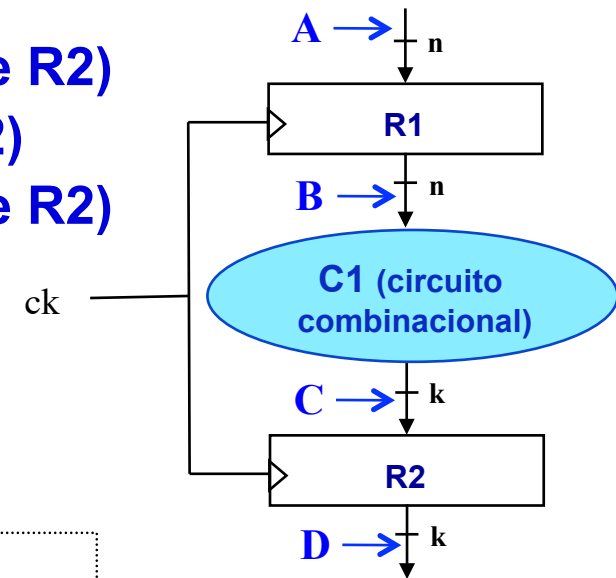
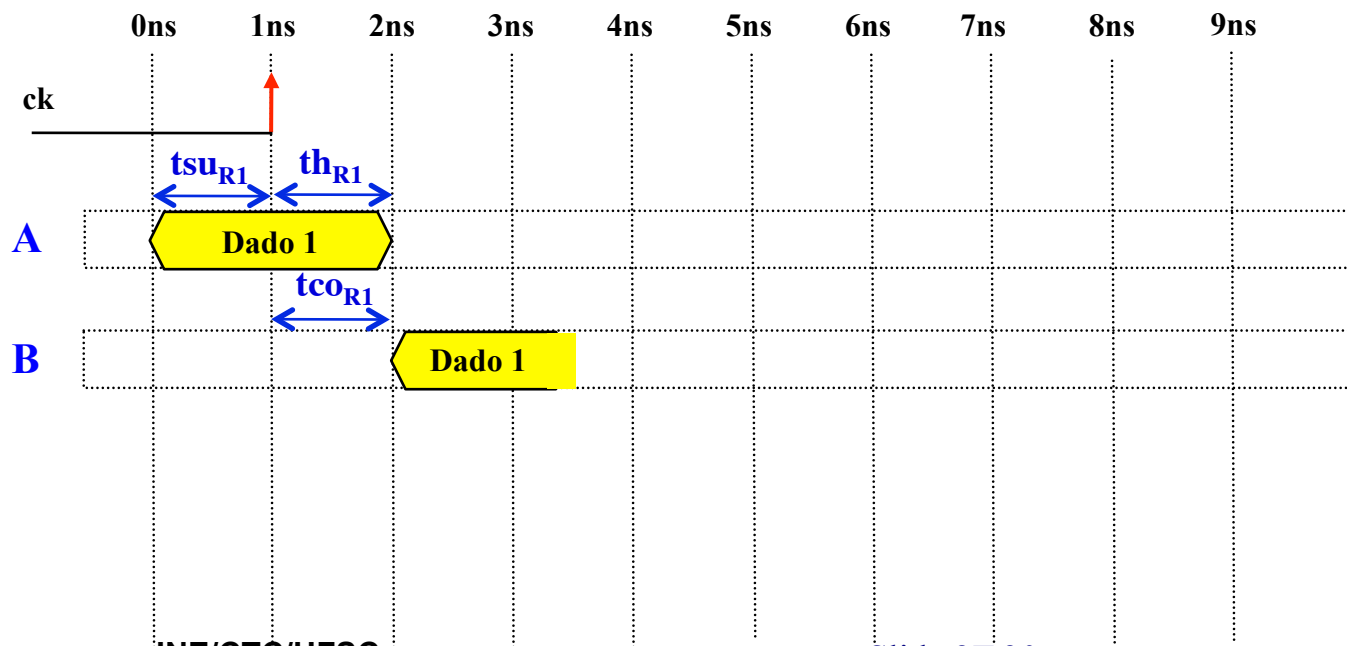


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)

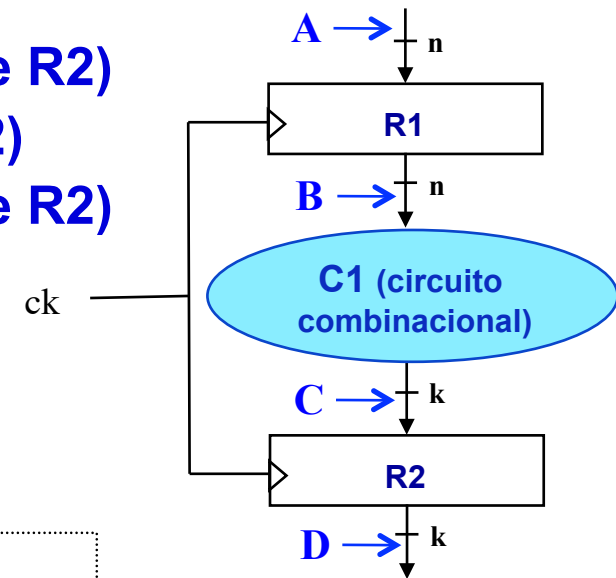
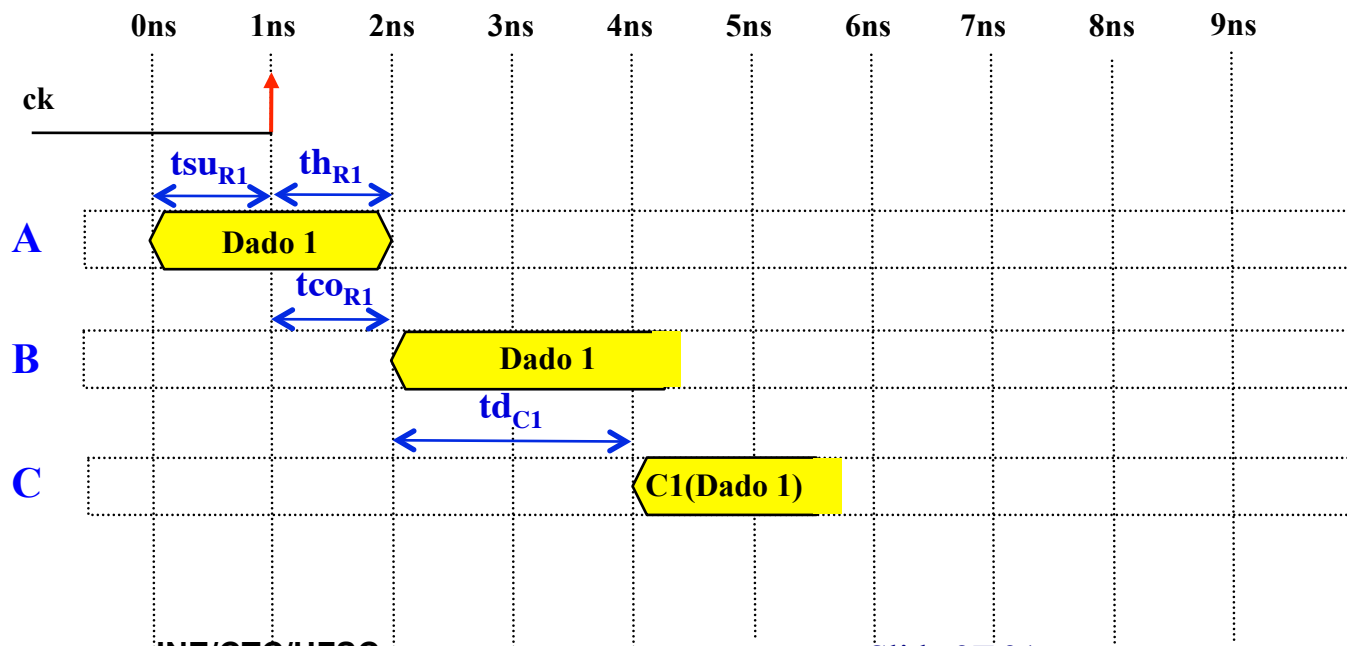


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $t_{su_{R1}} = t_{su_{R2}} = 1ns$ (tempo de setup de R1 e de R2)
- $t_{h_{R1}} = t_{h_{R2}} = 1ns$ (tempo de hold de R1 e de R2)
- $t_{co_{R1}} = t_{co_{R2}} = 1ns$ (tempo de carga de R1 e de R2)
- $t_{d_{C1}} = 2ns$ (atraso crítico (máximo) de C1)

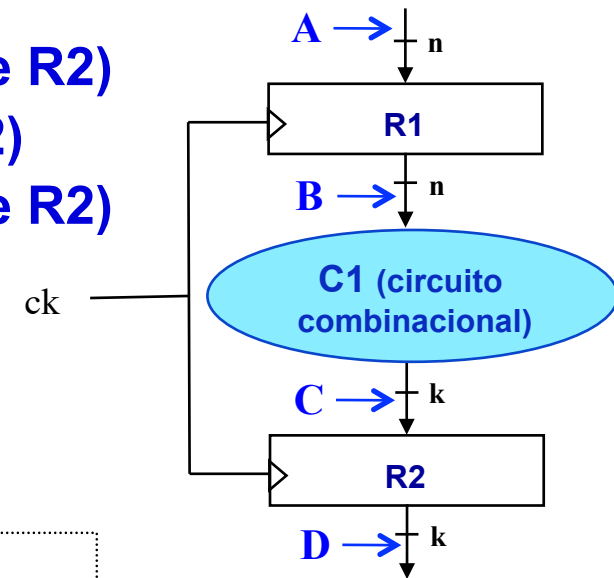
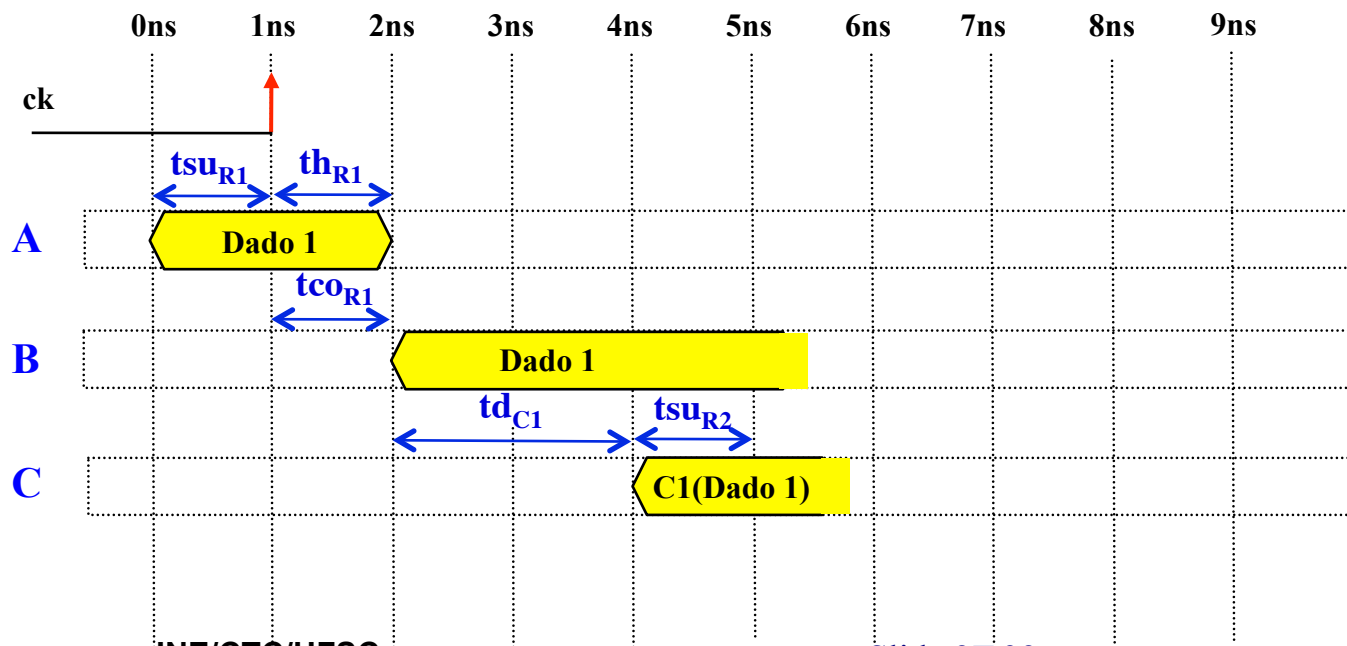


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)

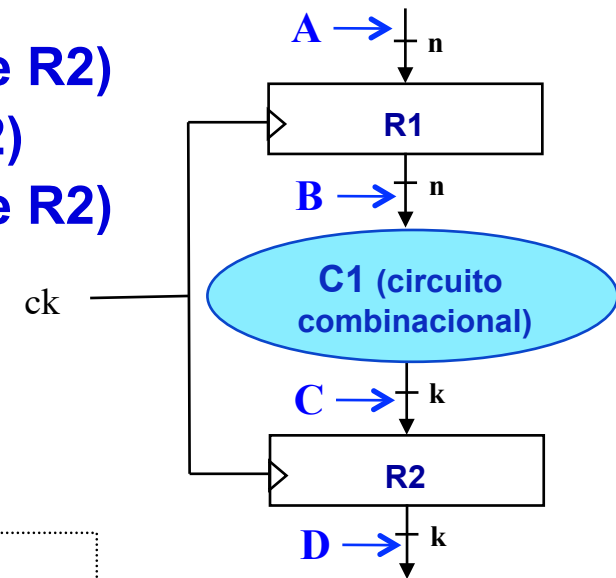
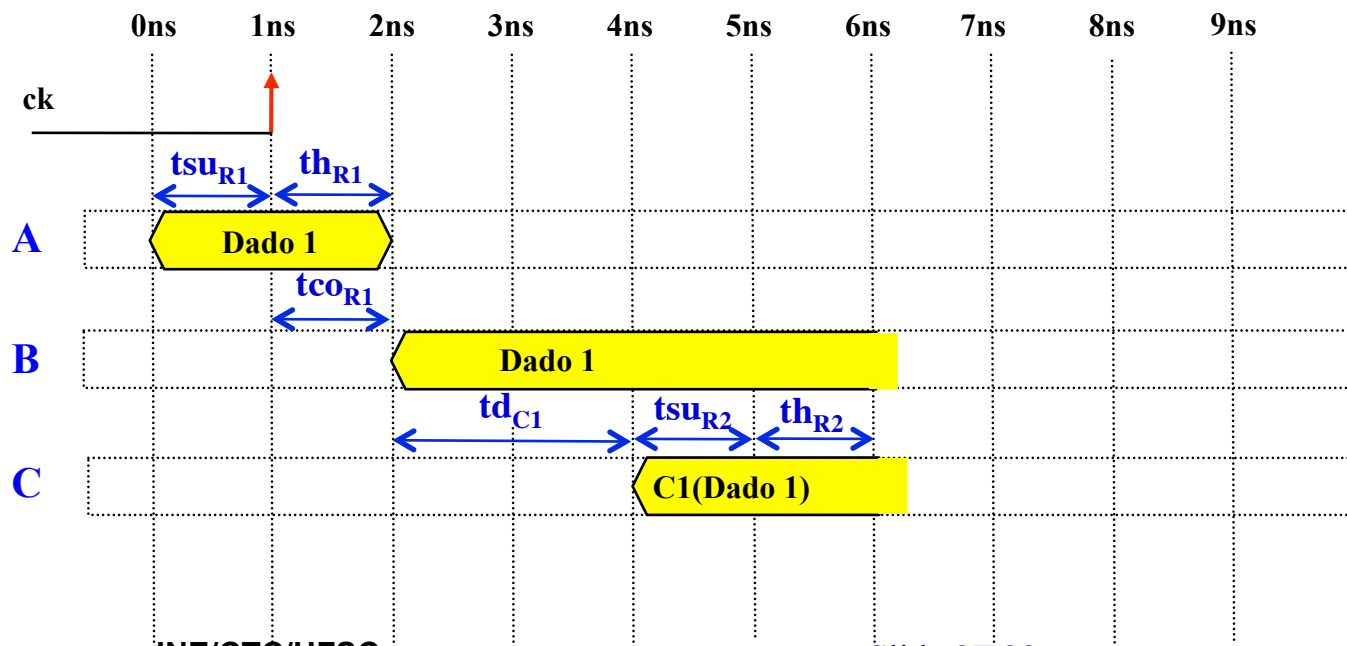


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)

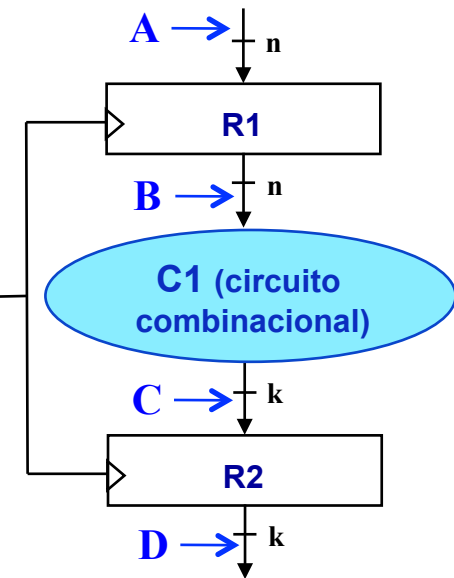
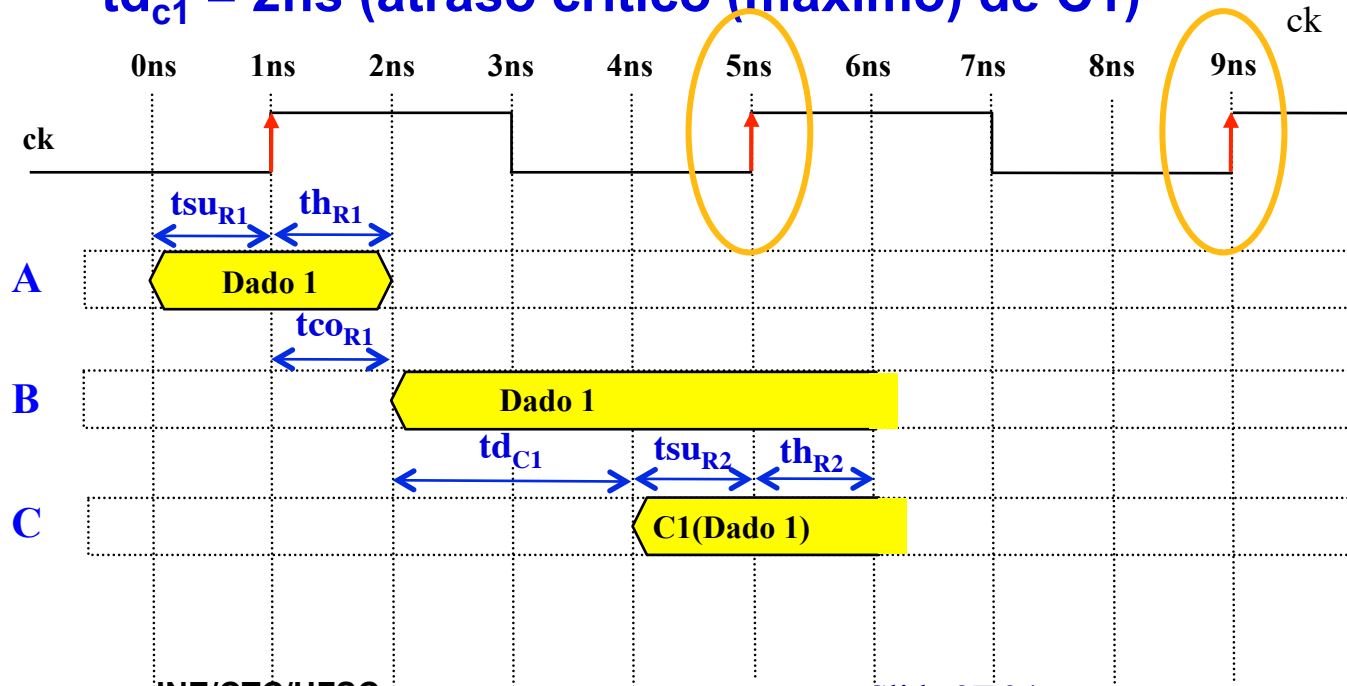


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)



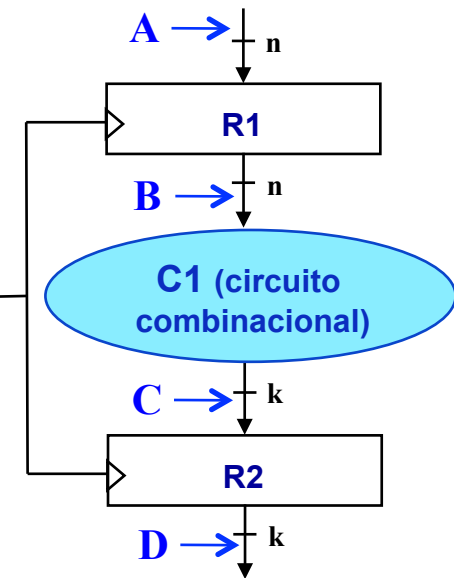
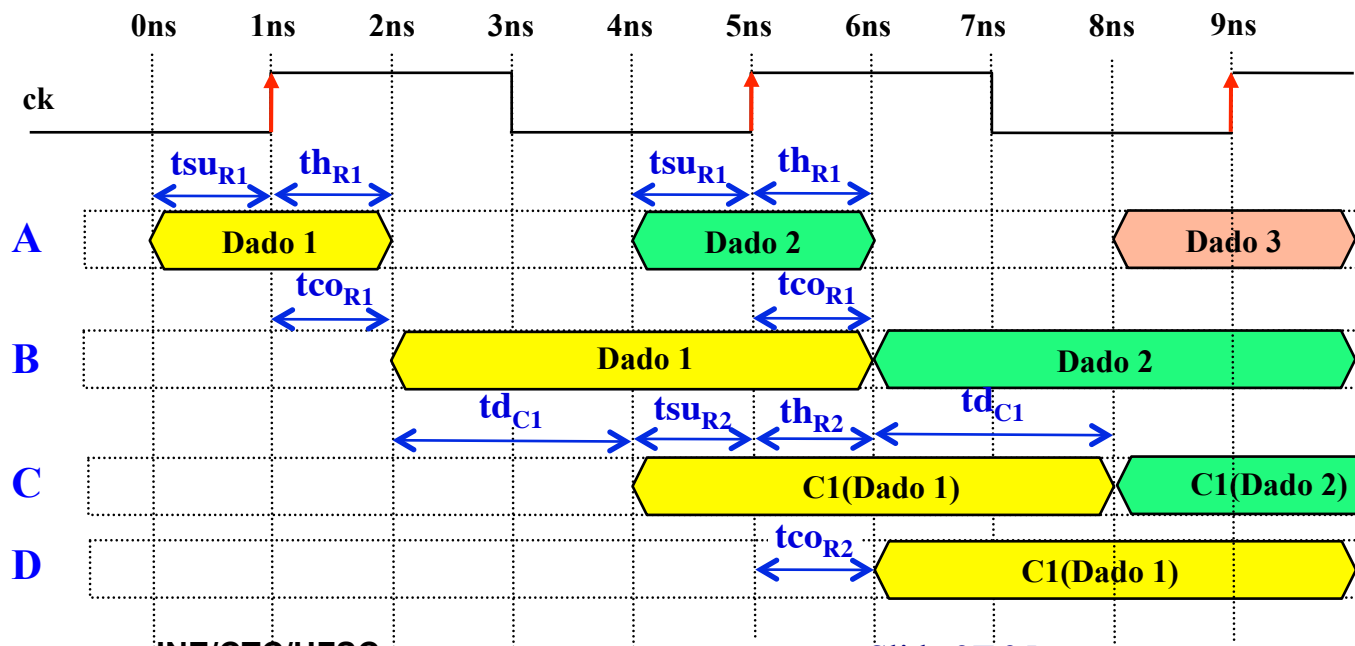
Período do relógio = 4ns

1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)

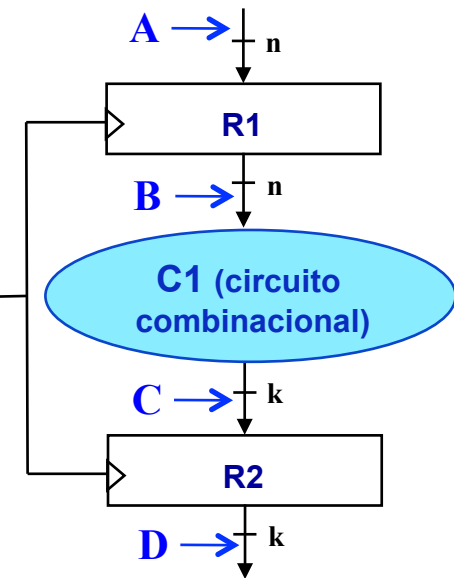
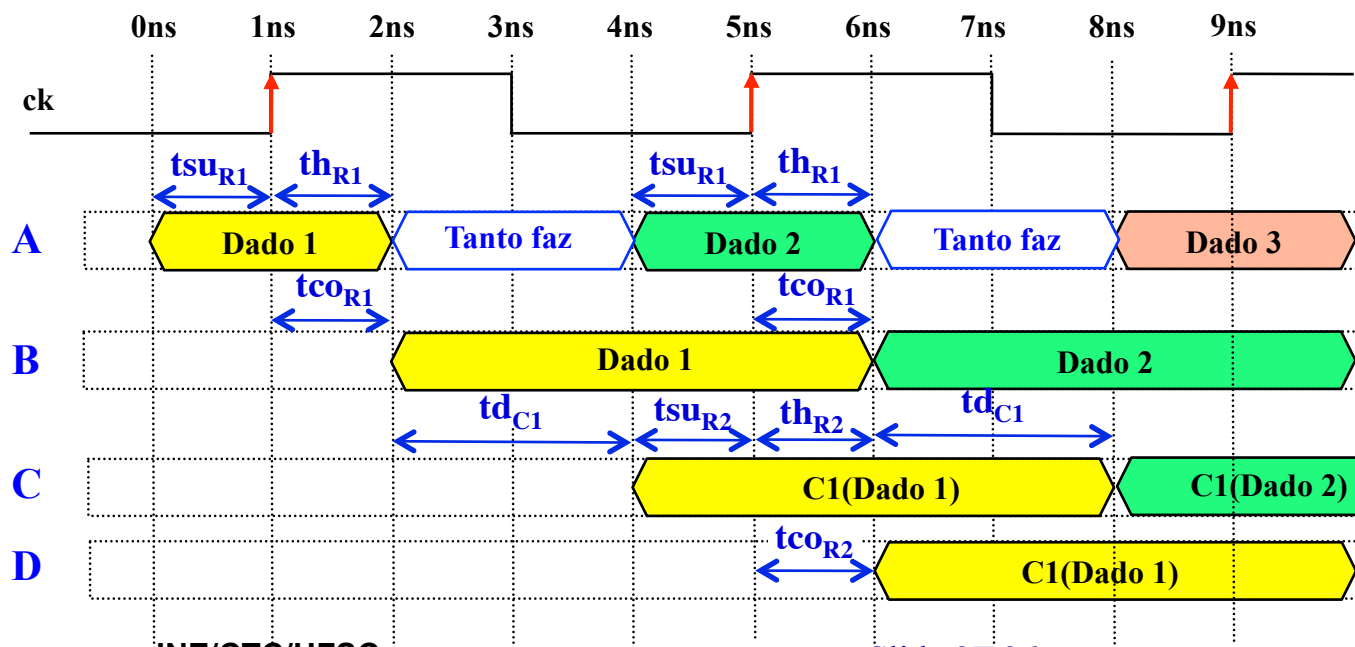


1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

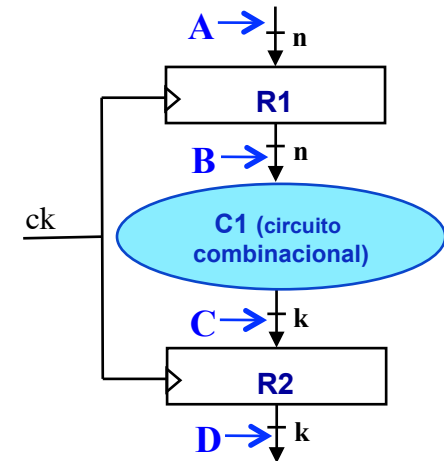
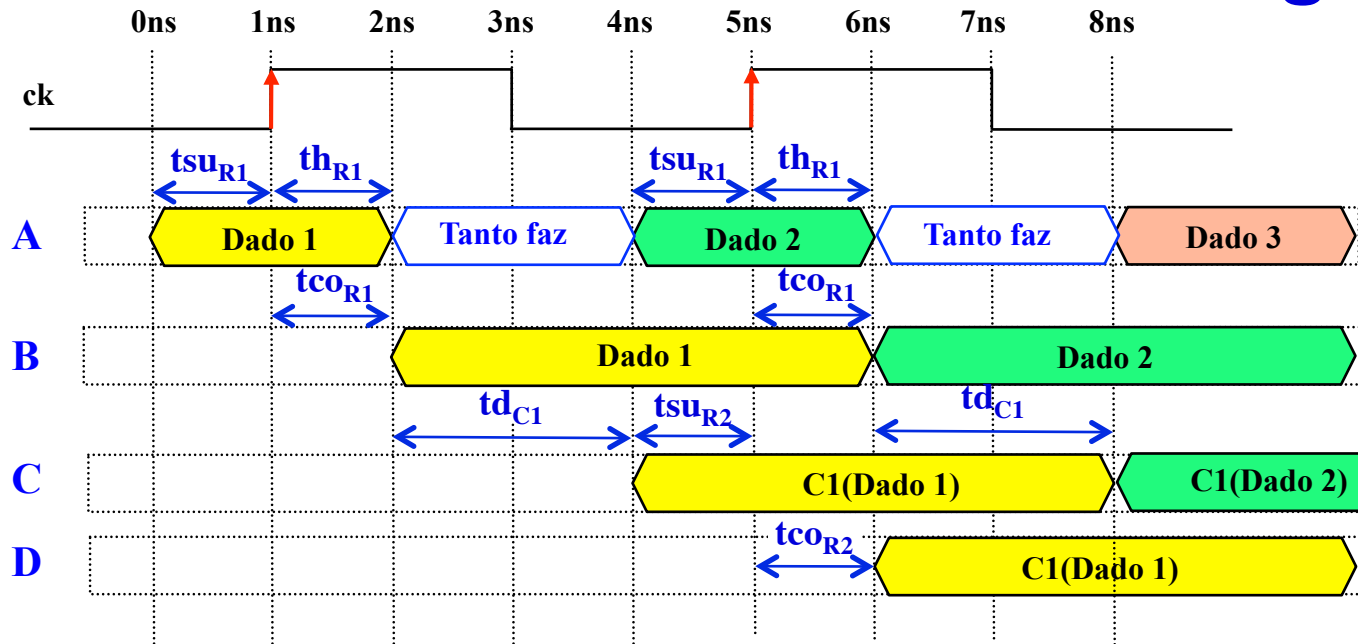
Supondo:

- $tsu_{R1} = tsu_{R2} = 1ns$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1ns$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1ns$ (tempo de carga de R1 e de R2)
- $td_{C1} = 2ns$ (atraso crítico (máximo) de C1)



1. Projeto de Unidade Lógico-Aritmética

▶ Cálculo do Período do Relógio



Conclusão. O período do relógio pode ser calculado por:

$$T = t_{co_{R1}} + t_{d_{C1}} + t_{su_{R2}} = 1 + 2 + 1 = 4 \text{ ns}$$

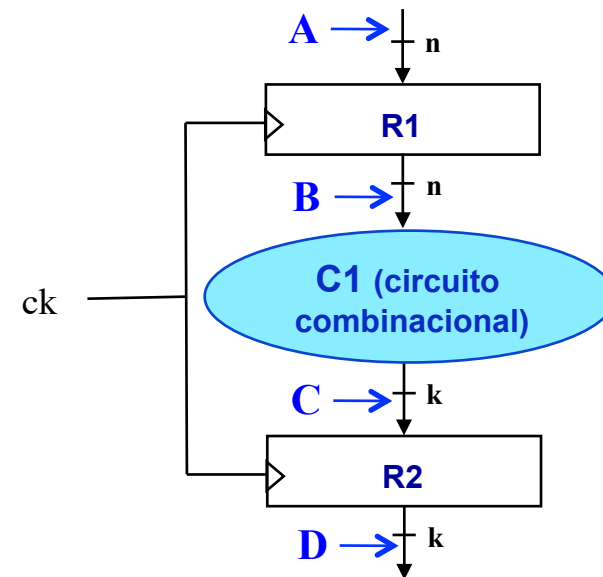
1. Projeto de Unidade Lógico-Aritmética

▶ Diagramas de Tempo

Pergunta:

Como estimar td_{C1} ?

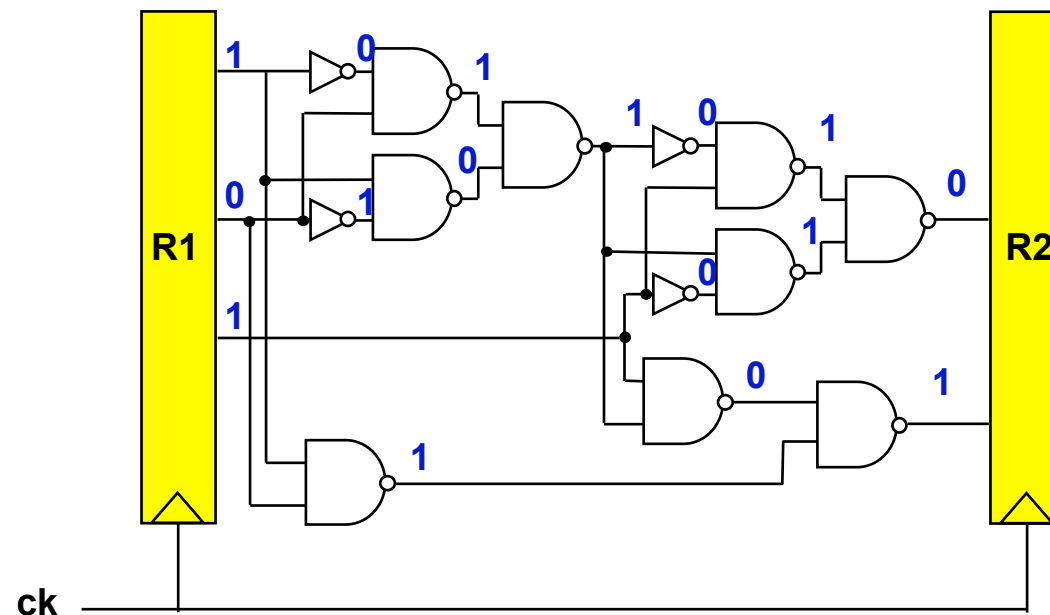
- Notar que C1 é um circuito combinacional



1. Projeto de Unidade Lógico-Aritmética

▶ Estimando o Atraso Crítico de Circuito Combinacional

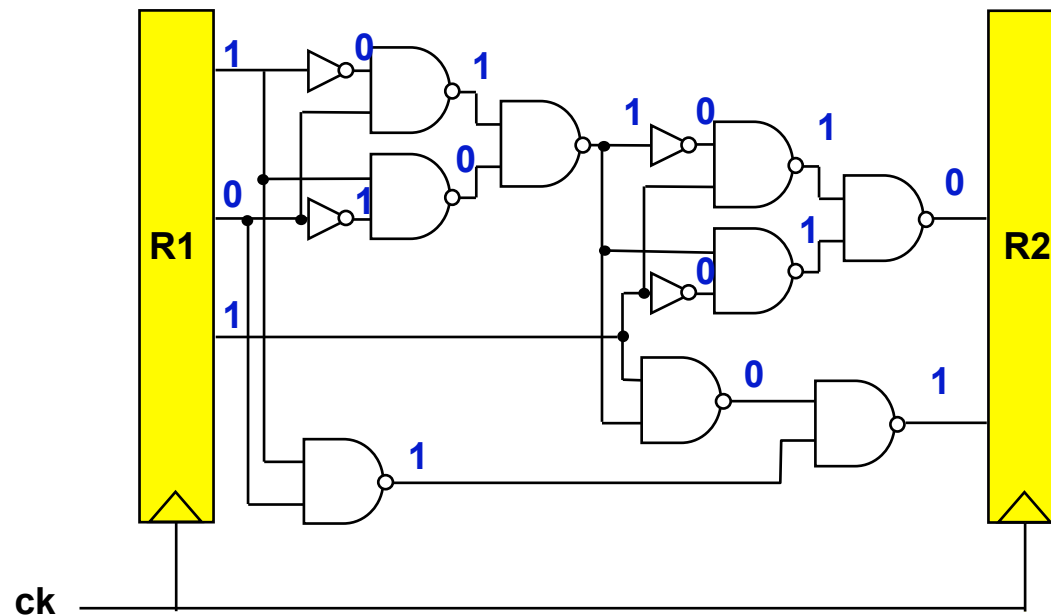
Supondo que $t_d=1$ unid. para qualquer porta lógica



1. Projeto de Unidade Lógico-Aritmética

▶ Estimando o Atraso Crítico de Circuito Combinacional

Supondo que $t_d=1$ unid. para qualquer porta lógica



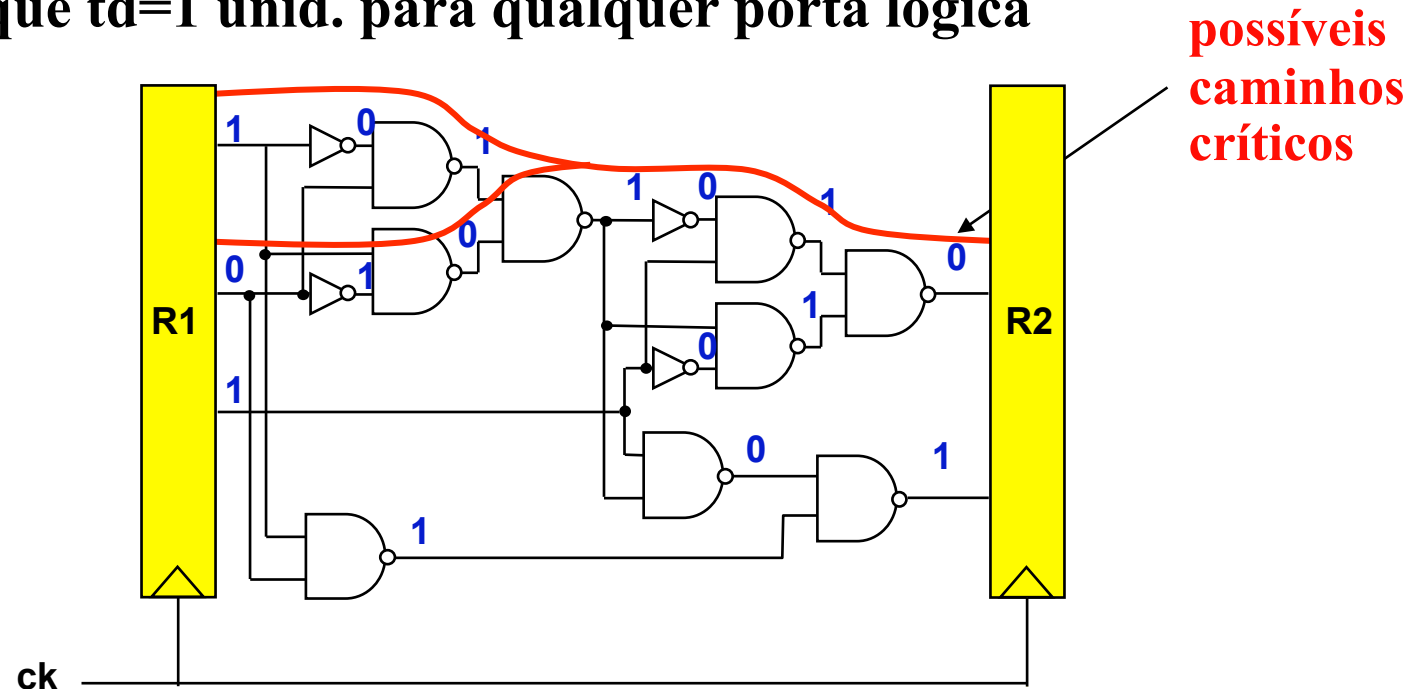
Possíveis Técnicas:

1. Simulação: repetir a análise para cada um dos 2^n vetores de entrada.

1. Projeto de Unidade Lógico-Aritmética

▶ Estimando o Atraso Crítico de Circuito Combinacional

Supondo que $t_d=1$ unid. para qualquer porta lógica



Possíveis Técnicas:

1. **Simulação:** repetir a análise para cada um dos 2^n vetores de entrada.
2. **Análise de Timing (STA):** analisar os atrasos dos caminhos entre R1 e R2.