MC542

Organização de Computadores
Teoria e Prática

2007

Prof. Paulo Cesar Centoducatte

ducatte@ic.unicamp.br

www.ic.unicamp.br/~ducatte

MC542

Circuitos Lógicos

Projeto de Circuitos Seqüenciais Síncronos

Máquinas de Estados Finitos

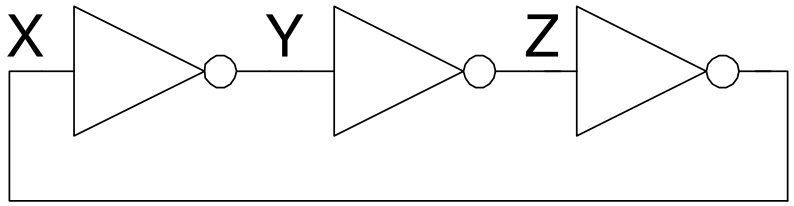
"DDCA" - (Capítulo 3)
"FDL" - (Capítulo 8)

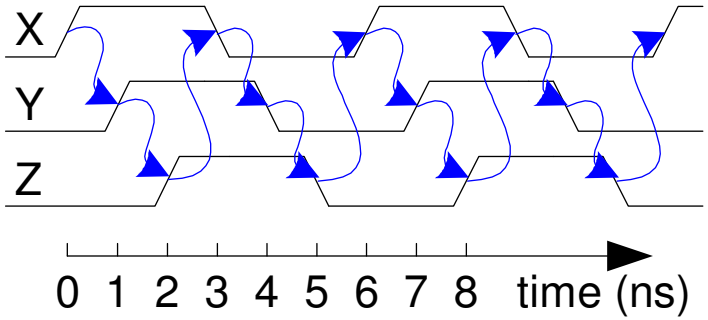
Título do Capítulo Abordado
Sumário

- Projeto de Circuitos lógicos Síncronos
- Projeto de Máquinas de Estados - FSM
  - Diagramas de Estados
  - Tabela de Estados
  - Atribuição de Estados
  - Escolha dos Flip-Flops
  - Derivação do Próximo Estado e Saída
- Exemplos
- Timing
  - Restrições de timing de entrada
  - Restrições de timing de saída

Lógica Seqüencial

- Circuito Seqüencial: todo aquele que não é um circuito combinacional
- Circuito problemático:





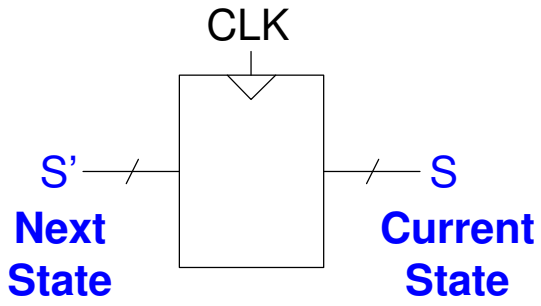
- Este circuito não tem entrada e tem de 1 a 3 saídas.
- Ele é um circuito instável que oscila.
- Seu período depende do atraso (delay) dos inversores – que por sua vez depende do processo de fabricação, dimensões, temperatura etc.
- O circuito possue um caminho ciclico (fedback)

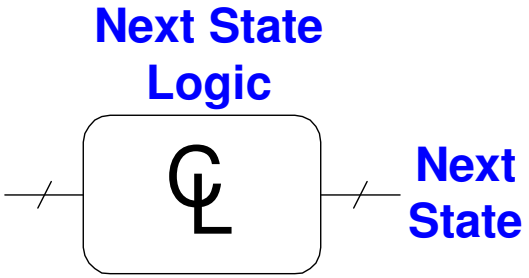
Projeto de Circuitos Seqüenciais Síncronos

- Quebre os caminhos ciclicos inserindo registradores
- Estes registradores conterão o estado do sistema
- O estado só muda na borda (de subida ou de descida) do clock
- Regras para composição de circuitos seqüenciasi síncronos:
  - Todo elemento do circuito ou é um registrador ou um circuito combincional
  - Há pelo menos um registrador.
  - Todos os registradores recebem o mesmo sinal de clock.
  - Todo caminho ciclico possue pelo menos um registrador.
- Exemplo de circuitos seqüenciais síncronos
  - Finite state machines (FSMs)
  - Pipelines

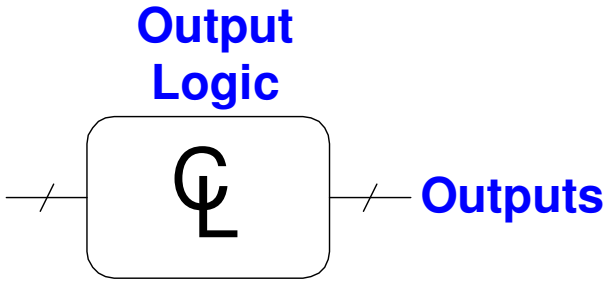
Finite State Machine (FSM)

- Constituido de :
  - Registradores de Estado que
    - » Armazenam o estado corrente e
    - » Carregam o próximo estado na borda do clock

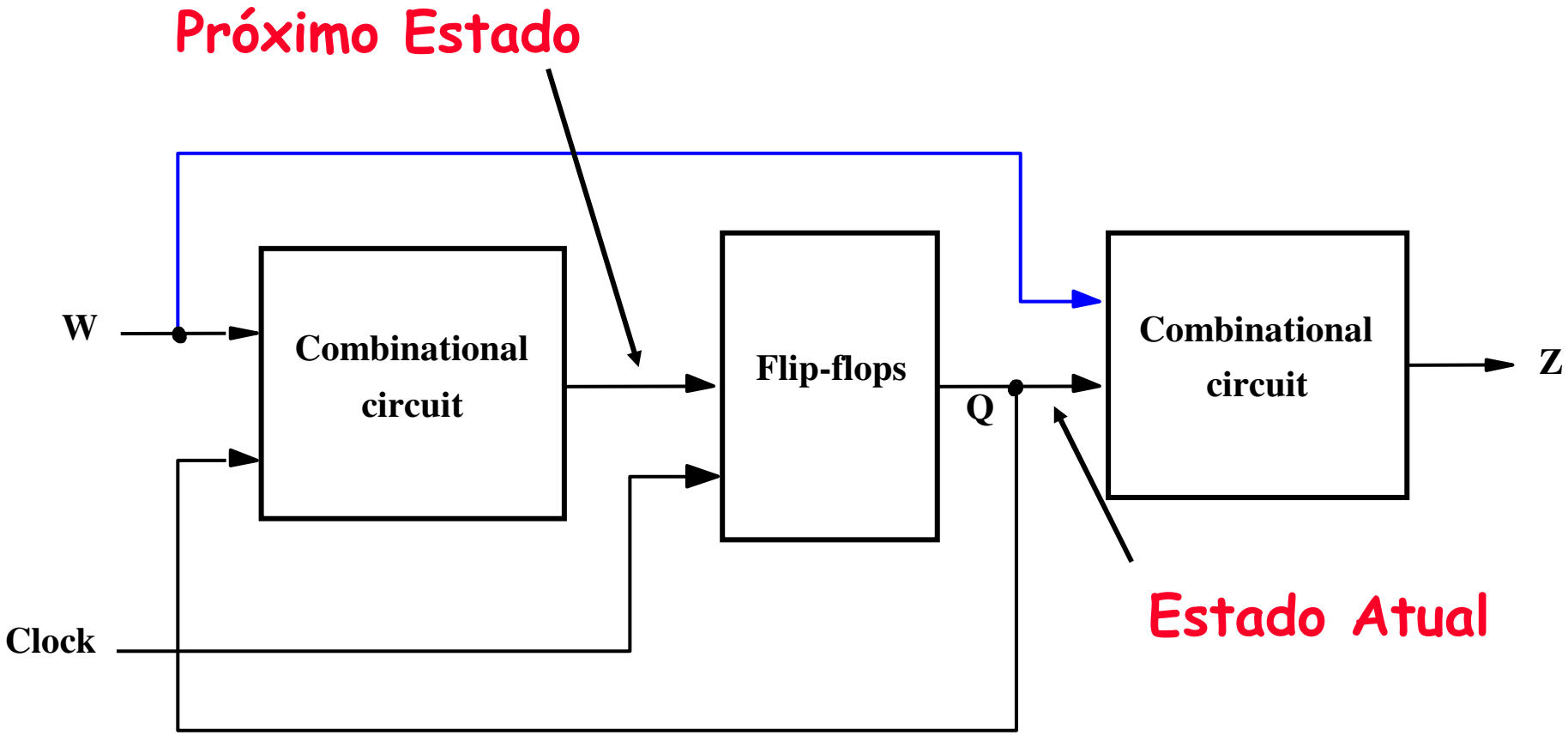




  - Circuito Combinacional que
    - » Computa o próximo estado
    - » Computa as saídas



Finite State Machine (FSM)



Máquina de Moore

Máquina de Mealy

Circuitos Seqüenciais Síncronos

- O circuito possui uma entrada w e uma saída s

- Toda mudança no circuito ocorre na borda do clock

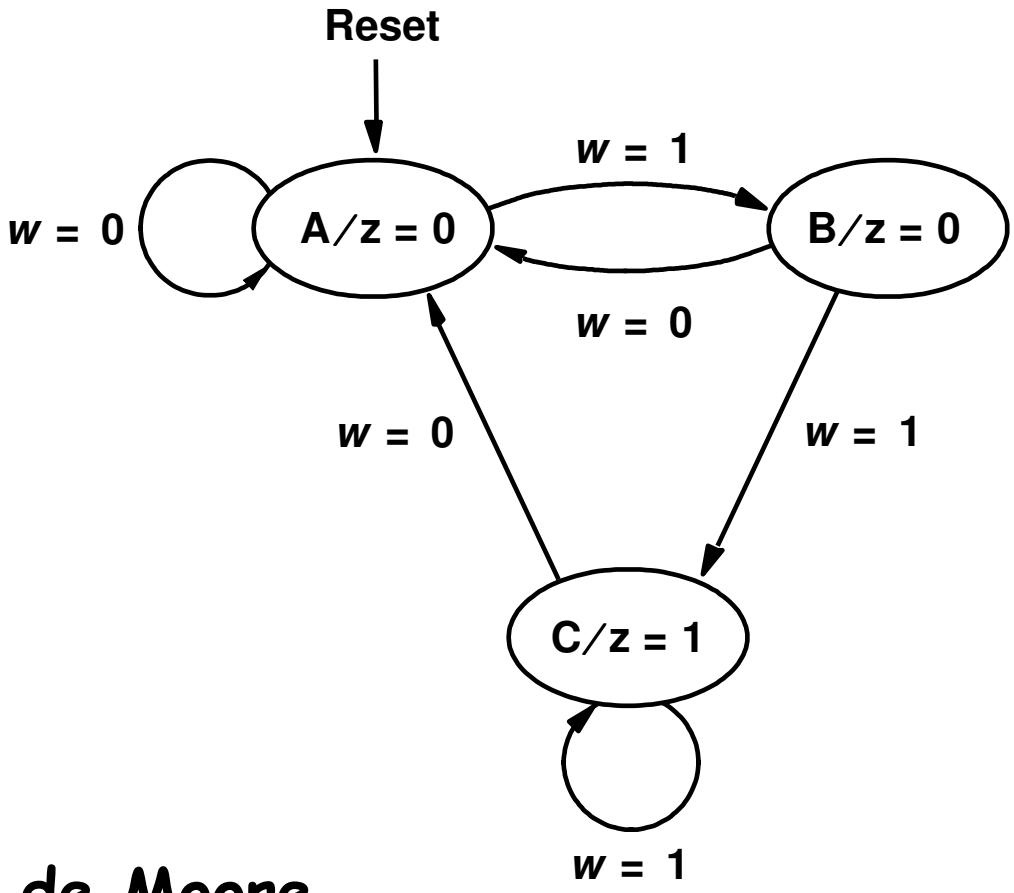
Exemplo: Projetar um circuito que possui uma entrada w de 1 bit e uma saída z também de 1 bit e z = 1 se w = 1 durante os dois ciclos de clock precedentes e z = 0 caso contrário. O circuito deve responder à borda de subida do clock.

Circuitos Seqüenciais Síncronos

- Exemplo do comportamento do circuito a ser projetado

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Diagrama de Estados

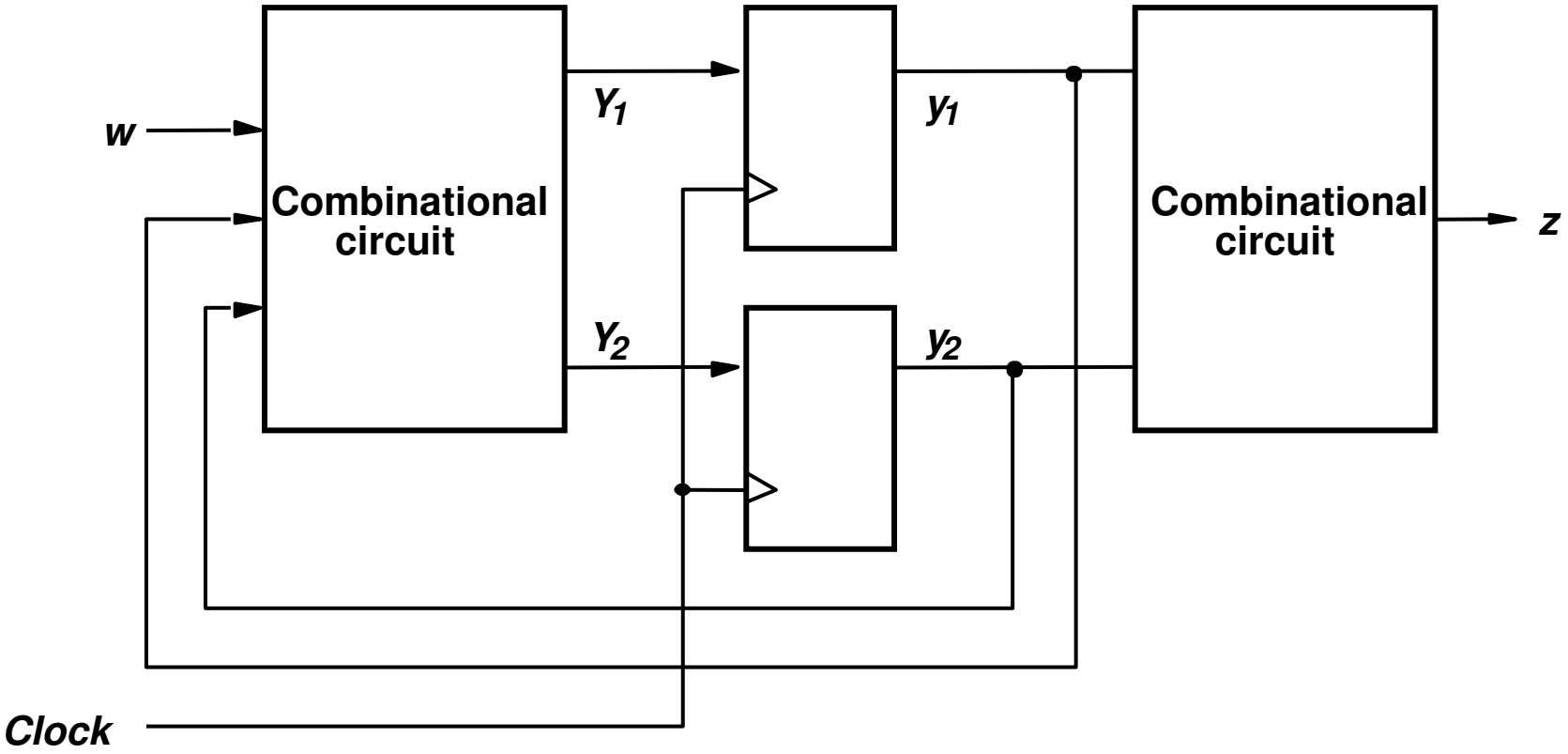


Máquina de Moore

Tabela de Estados

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1

Atribuição de Estados

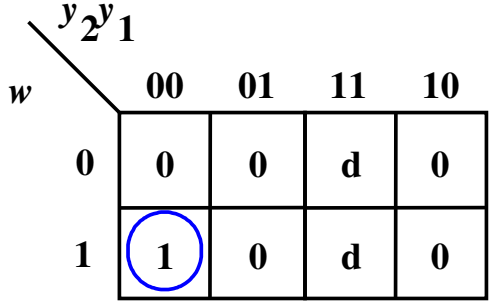


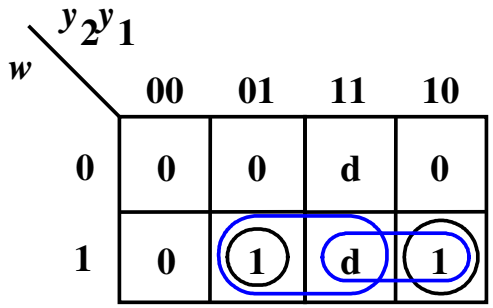
Atribuição de Estados

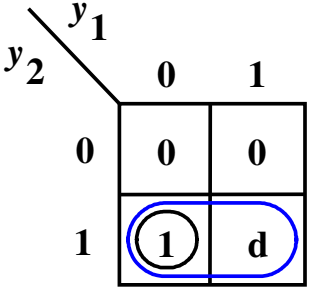
	Present state $y_2 y_1$	Next state $w = 0$ $Y_2 Y_1$	Next state $w = 1$ $Y_2 Y_1$	Output z
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	*dd*	*dd*	*d*

Esccolha dos Flip-Flops e Derivação das Equações de Exitação e de Saída

FF tipo D







Ignoring don't cares	Using don't cares
$Y_1 = w\bar{y}_1\bar{y}_2$	$Y_1 = w\bar{y}_1\bar{y}_2$
$Y_2 = wy_1\bar{y}_2 + w\bar{y}_1y_2$	$Y_2 = wy_1 + wy_2$ $= w(y_1 + y_2)$
$z = \bar{y}_1y_2$	$z = y_2$

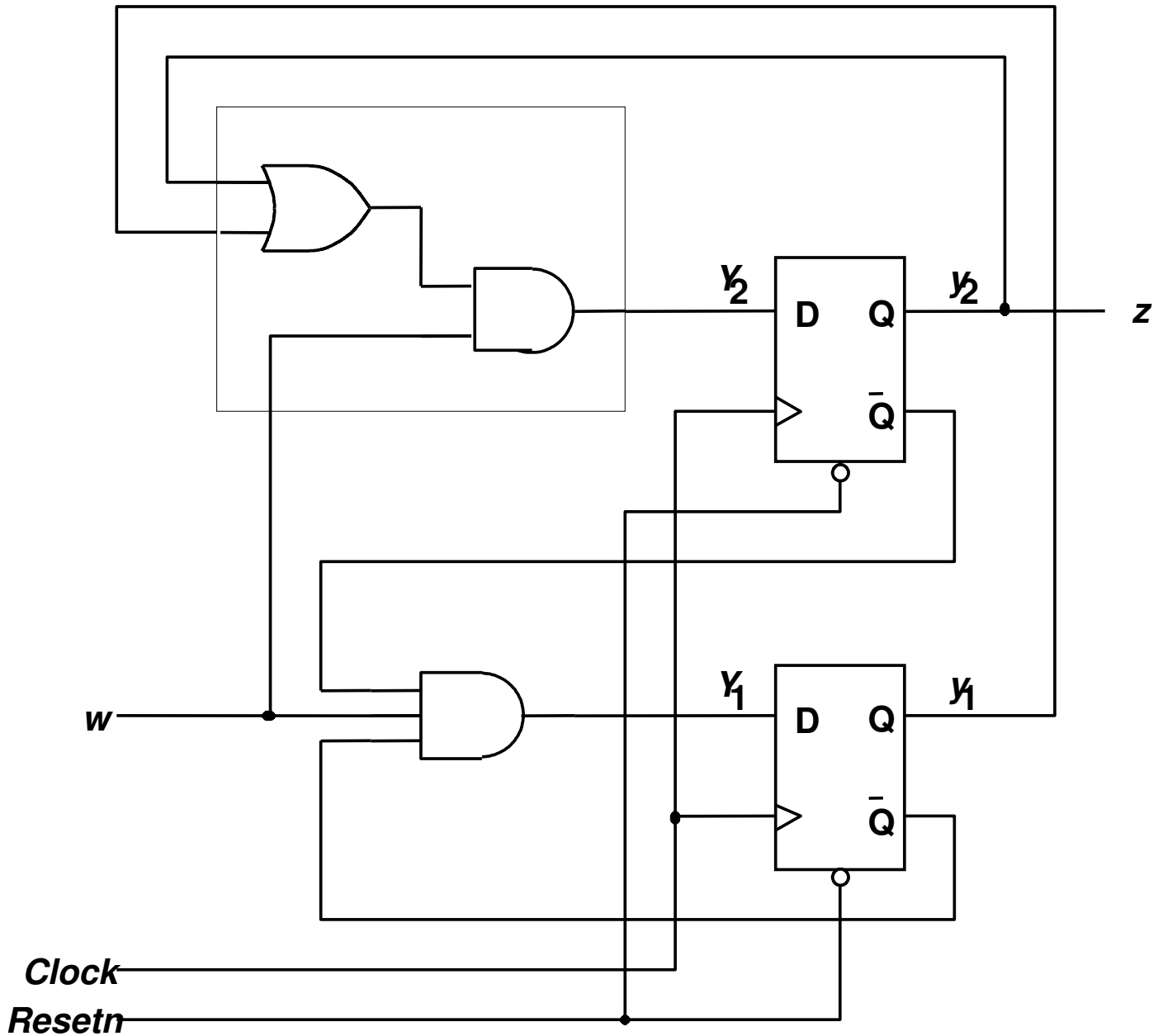
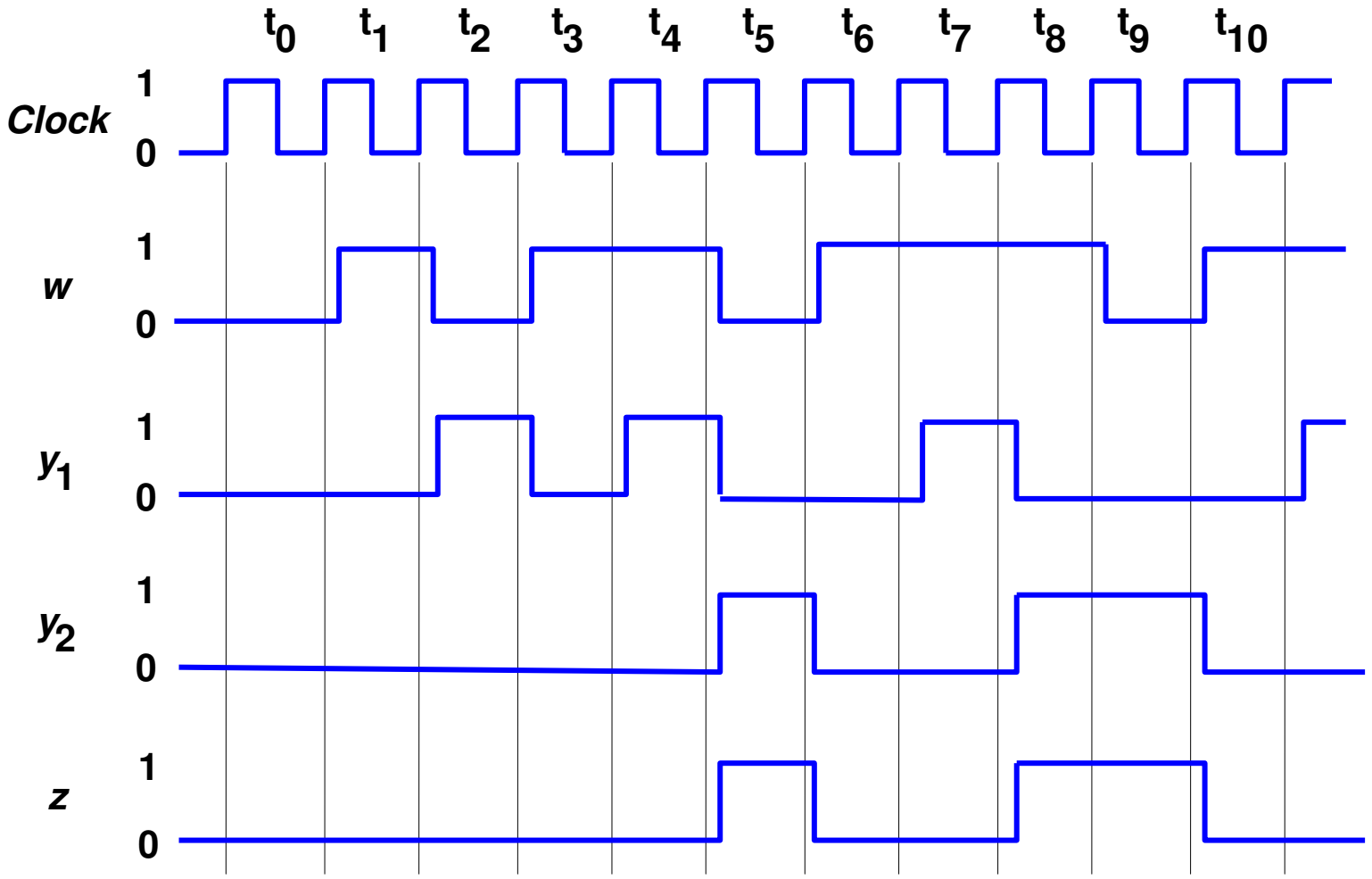
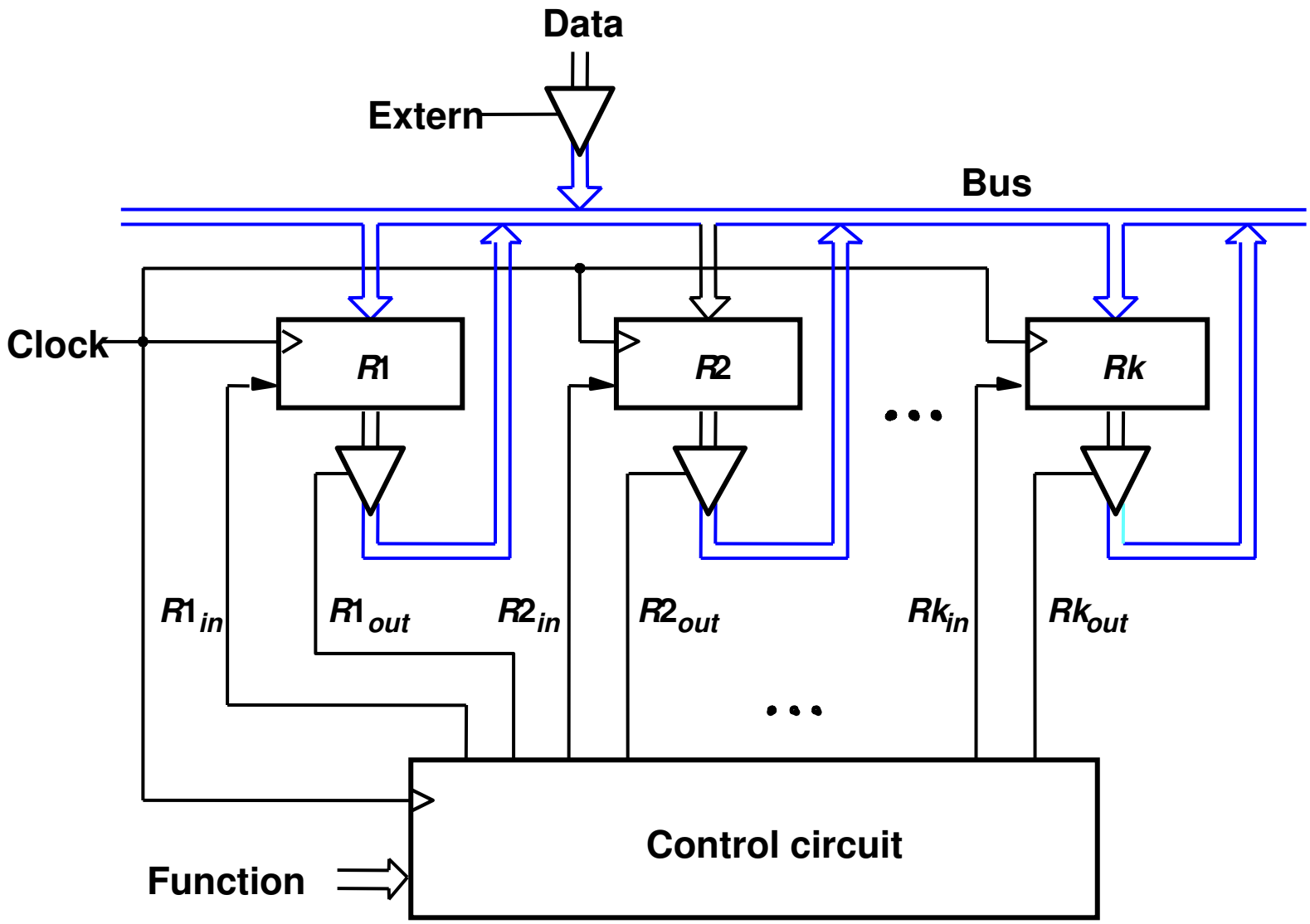

Y_2
D
Q
y_2
z
$\bar{Q}$
Y_1
D
Q
y_1
w
$\bar{Q}$
Clock
Resetn

Diagrama de Tempo

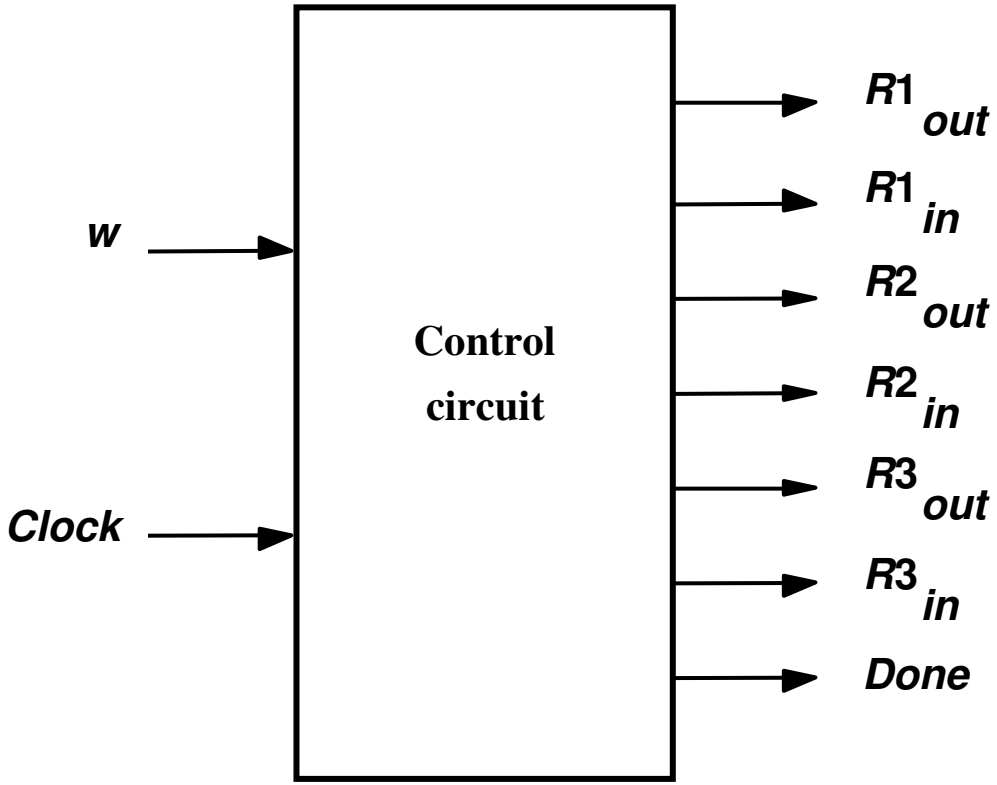


Exemplo: Registradores em um Barramento

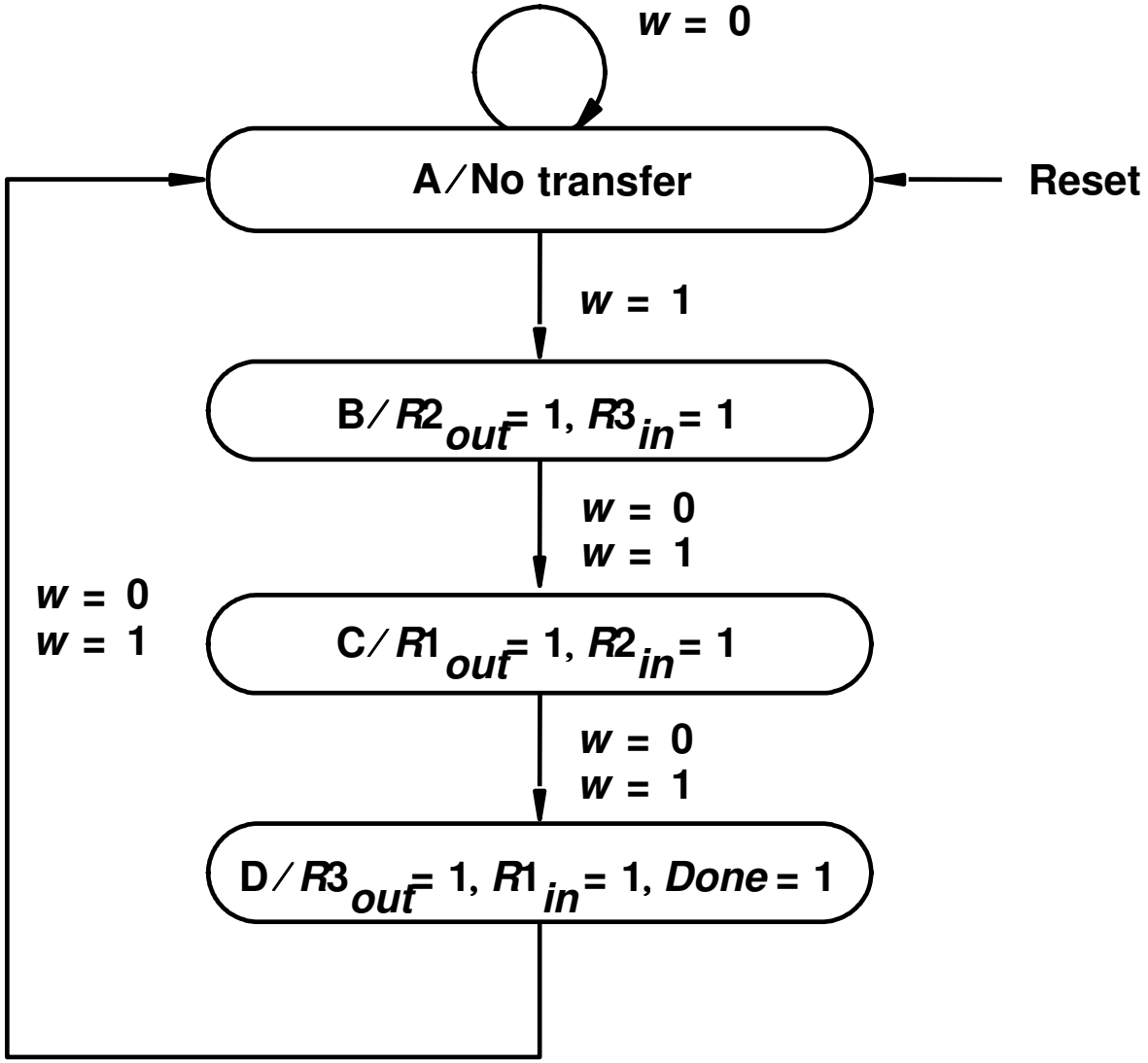


Exemplo (Cont.)

Projetar o Controle para realizar swap entre R1 e R2, usando R3 como auxiliar



Exemplo (Cont.)
Diagrama de Estados



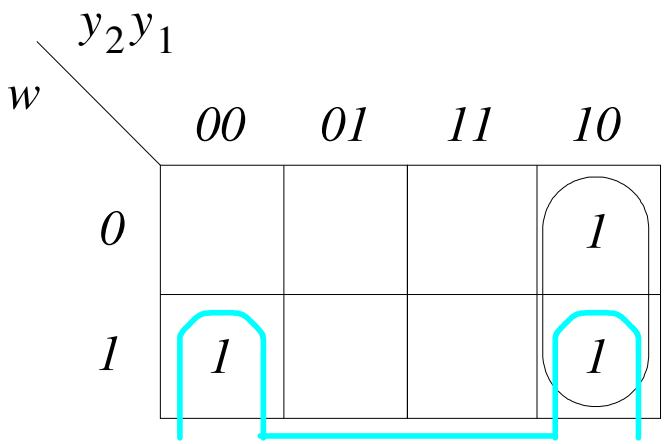
Exemplo (Cont.)
Tabela de Estados

Present state	Next state		Outputs						
	w = 0	*w* = 1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	$Done$
A	A	B	0	0	0	0	0	0	0
B	C	C	0	0	1	0	0	1	0
C	D	D	1	0	0	1	0	0	0
D	A	A	0	1	0	0	1	0	1

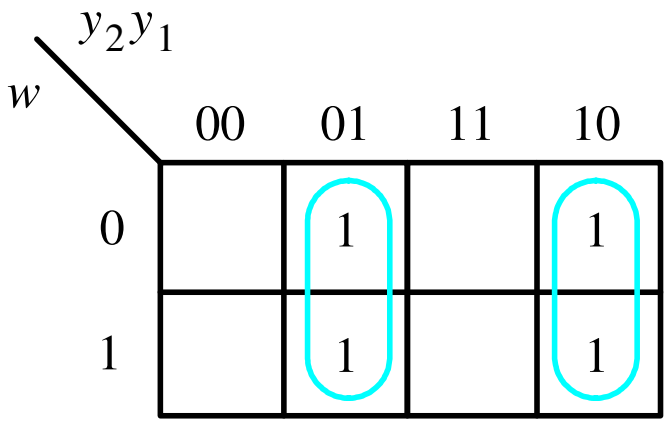
Exemplo (Cont.)
Tabela de Atribuição de Estados

	Present state	Nextstate		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	$Done$
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

Derivação das Equações de Exitação, para FF tipo D, e de Saída

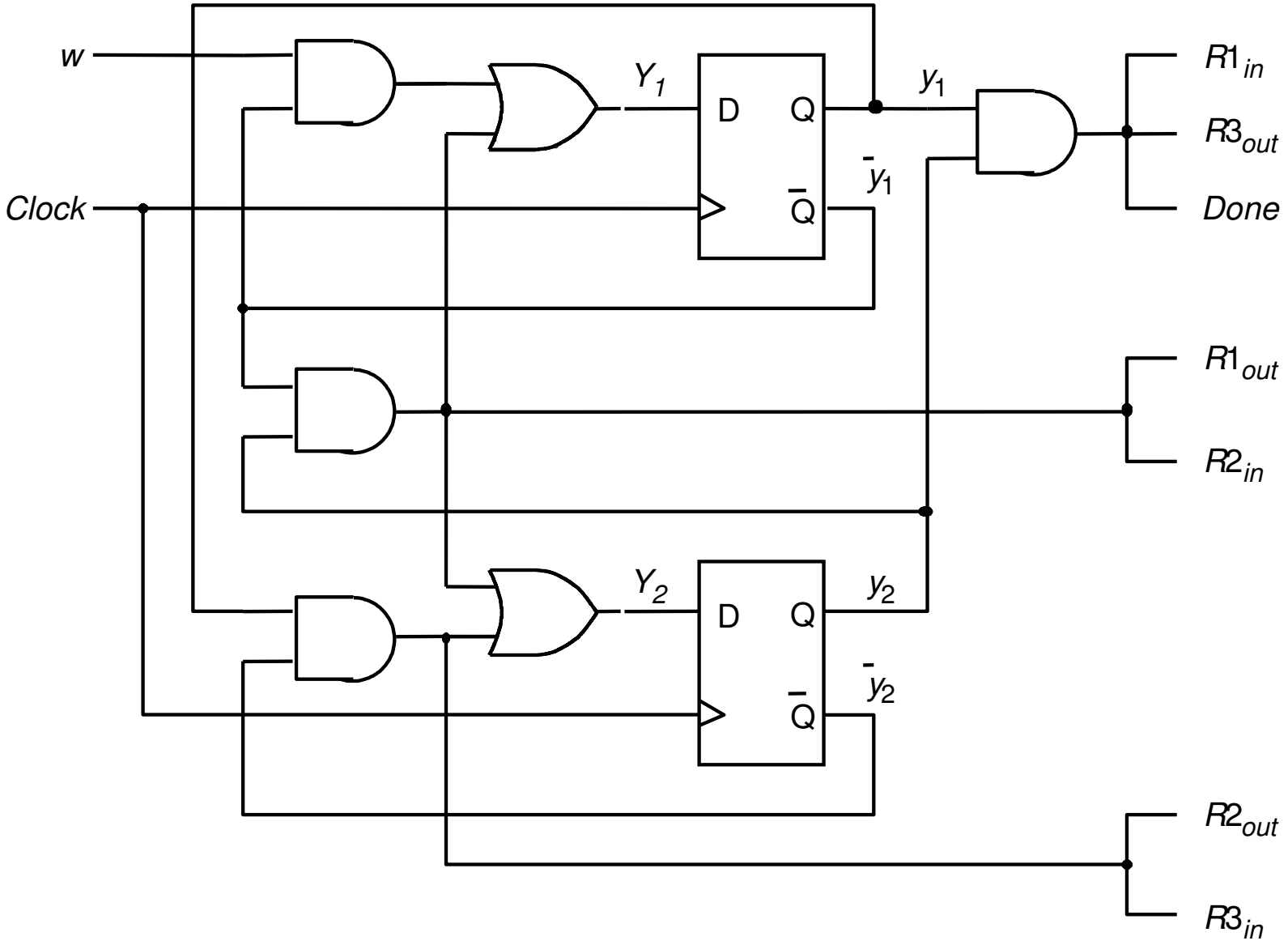


$$Y_1 = w\bar{y}_1 + \bar{y}_1 y_2$$



$$Y_2 = y_1\bar{y}_2 + \bar{y}_1 y_2$$

Derivação das Equações de Exitação e de Saída

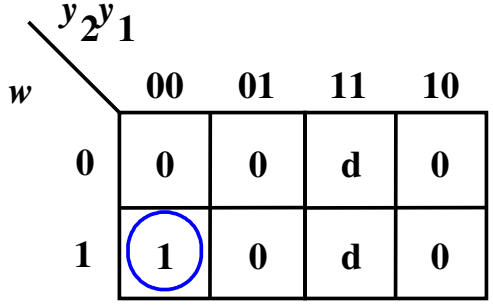


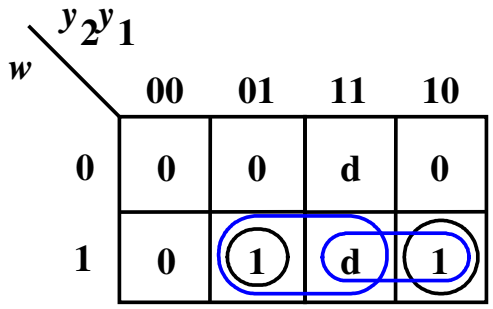
Atribuição de Estados

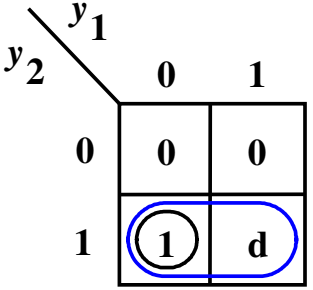
	Present state $y_2 y_1$	Next state $w = 0$ $Y_2 Y_1$	Next state $w = 1$ $Y_2 Y_1$	Output z
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	*dd*	*dd*	*d*

Esccolha dos Flip-Flops e Derivação das Equações de Exitação e de Saída

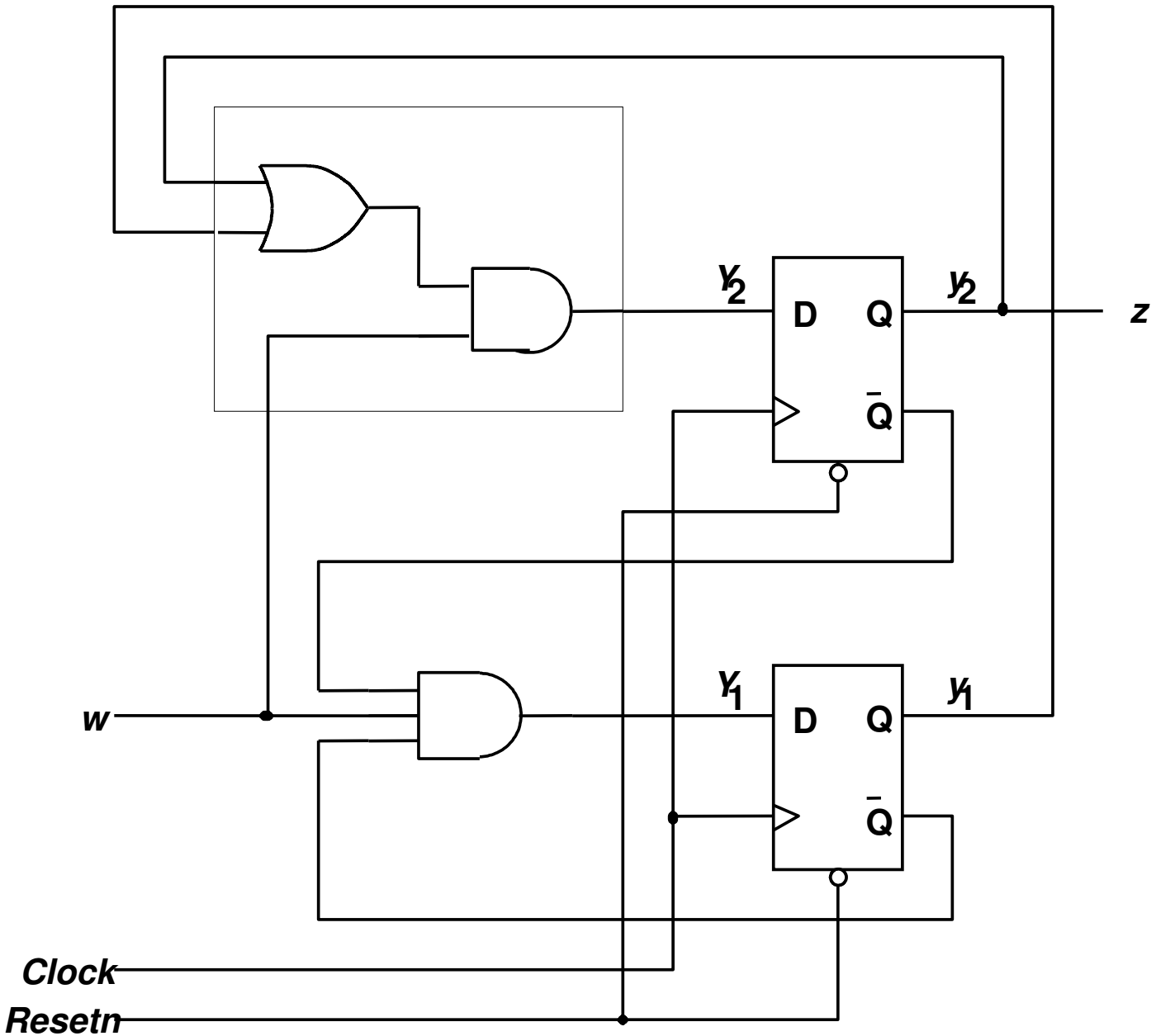
FF tipo D







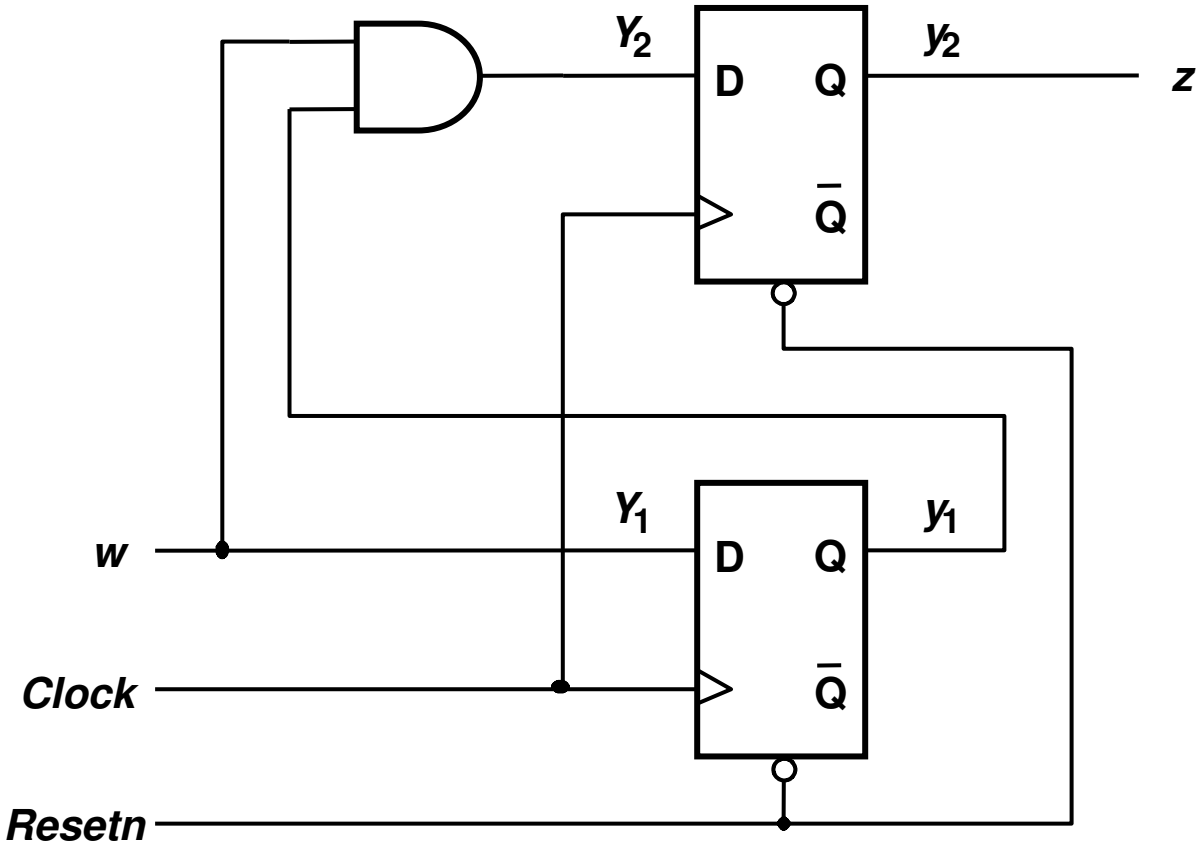
Ignoring don't cares	Using don't cares
$Y_1 = w\bar{y}_1\bar{y}_2$	$Y_1 = w\bar{y}_1\bar{y}_2$
$Y_2 = wy_1\bar{y}_2 + w\bar{y}_1y_2$	$Y_2 = wy_1 + wy_2$ $= w(y_1 + y_2)$
$z = \bar{y}_1y_2$	$z = y_2$


Y_2
D
Q
y_2
z
$\bar{Q}$
Y_1
D
Q
y_1
w
$\bar{Q}$
Clock
Resetn

Atribuição de Estados
Existe uma Solução Melhor?

	Present state	Next state		Output
		$w = 0$	$w = 1$	
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	z
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	*dd*	*dd*	*d*

Atribuição de Estados
Existe uma Solução Melhor?



Atribuição de Estados

- Para Circuitos grandes, diferentes Atribuições de Estados, tem um impacto considerável sobre o custo

- Na prática é impossível determinar a melhor atribuição de estados para circuitos grandes.

- Soluções com abordagem baseada em busca exaustiva são impraticáveis.

- Ferramentas de CAD usam, em geral, técnicas baseadas em heurísticas para realizar a atribuição de estados e os detalhes não são, em geral, publicados.

Atribuição de Estados
One-Hot Encoding

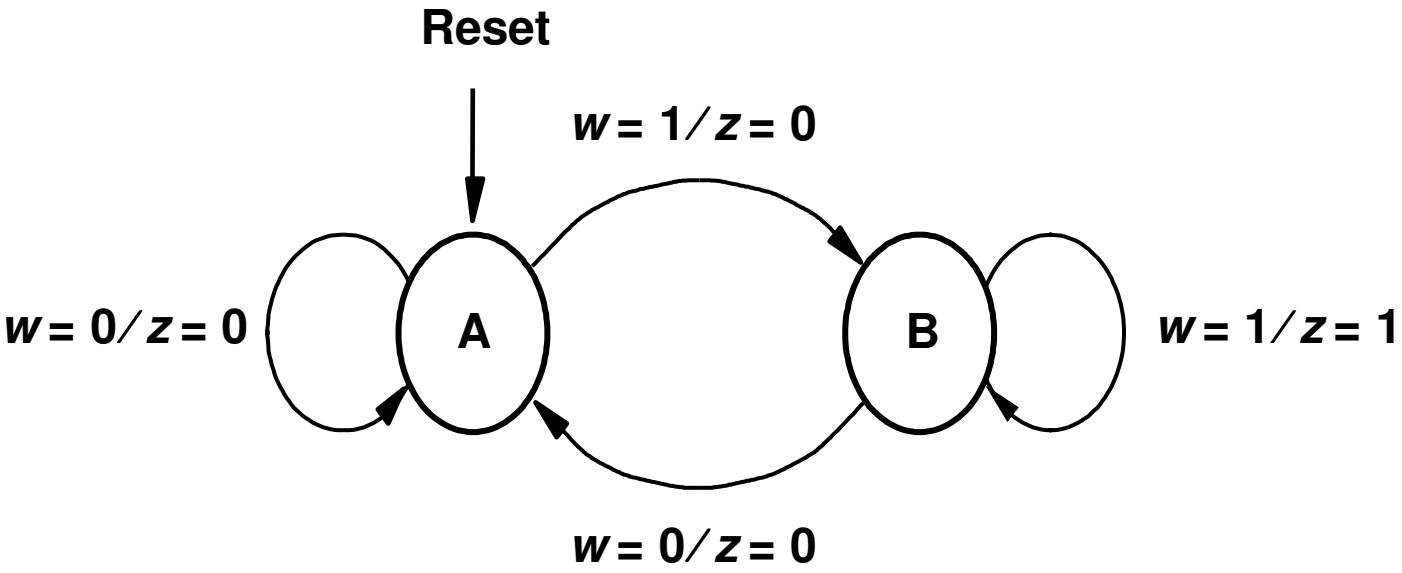
- Uso de tantas variáveis de Estados quantos forem os Estados.
- Um estado é representado com uma váriavel igual a 1 e todas as outras em 0
  - A variável igual a 1 é chamada de "hot"
- Exemplo:

	Present state	Nextstate		Output
		$w = 0$	$w = 1$	
	$y_3y_2y_1$	$Y_3Y_2Y_1$	$Y_3Y_2Y_1$	z
A	001	001	010	0
B	010	001	100	0
C	100	001	100	1

FSM de Mealy

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	1	0	0	1	1	0	0

FSM de Mealy



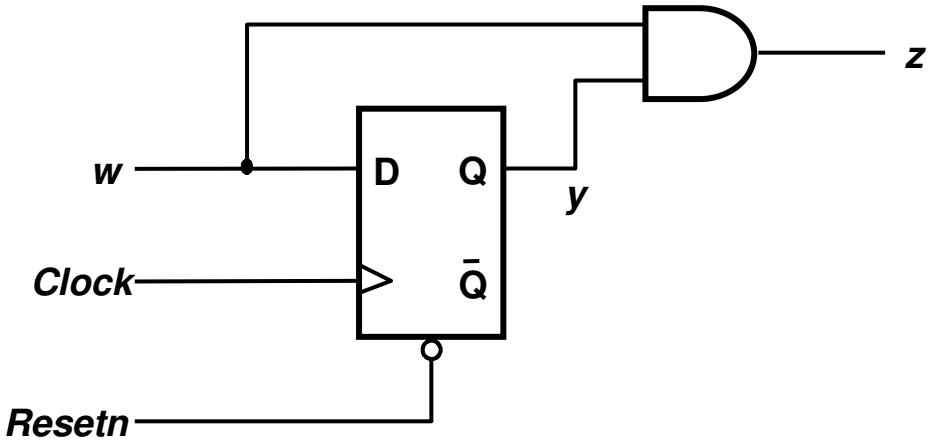
FSM de Mealy

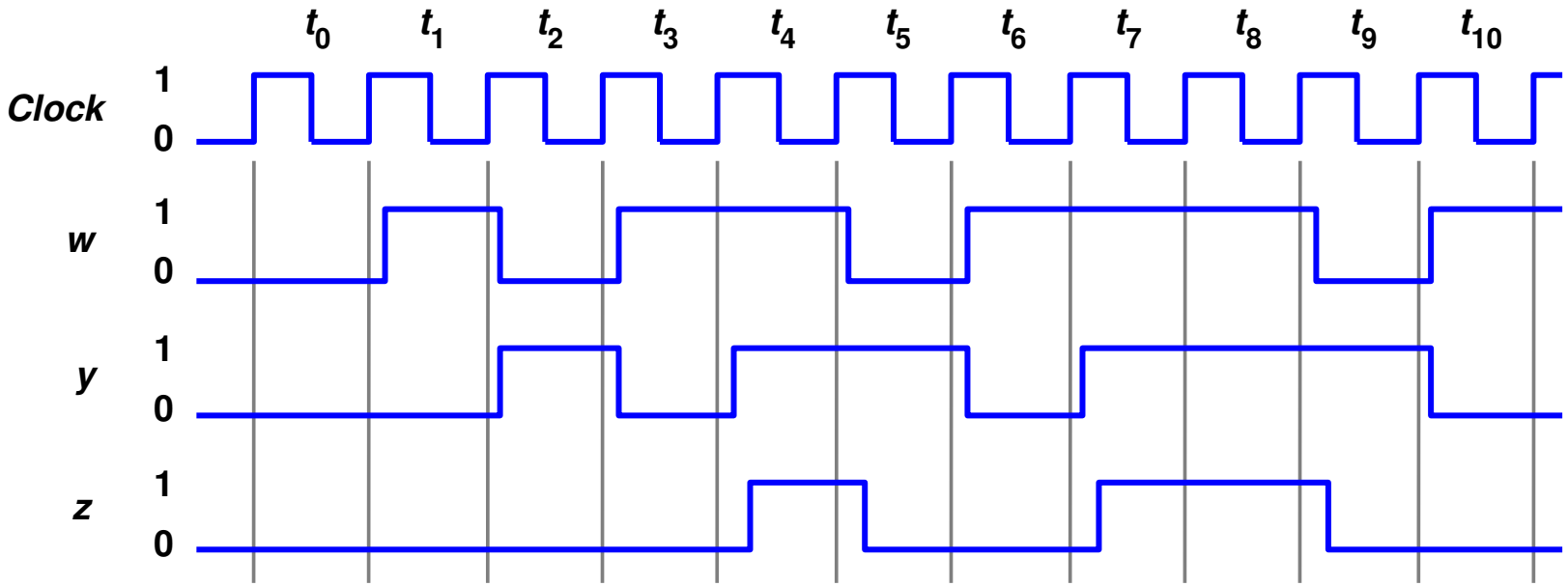
Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

FSM de Mealy

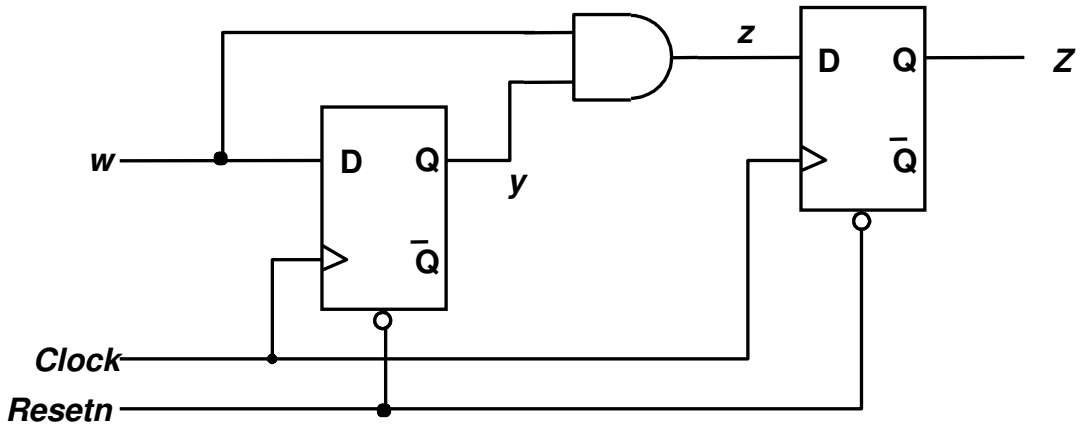
	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

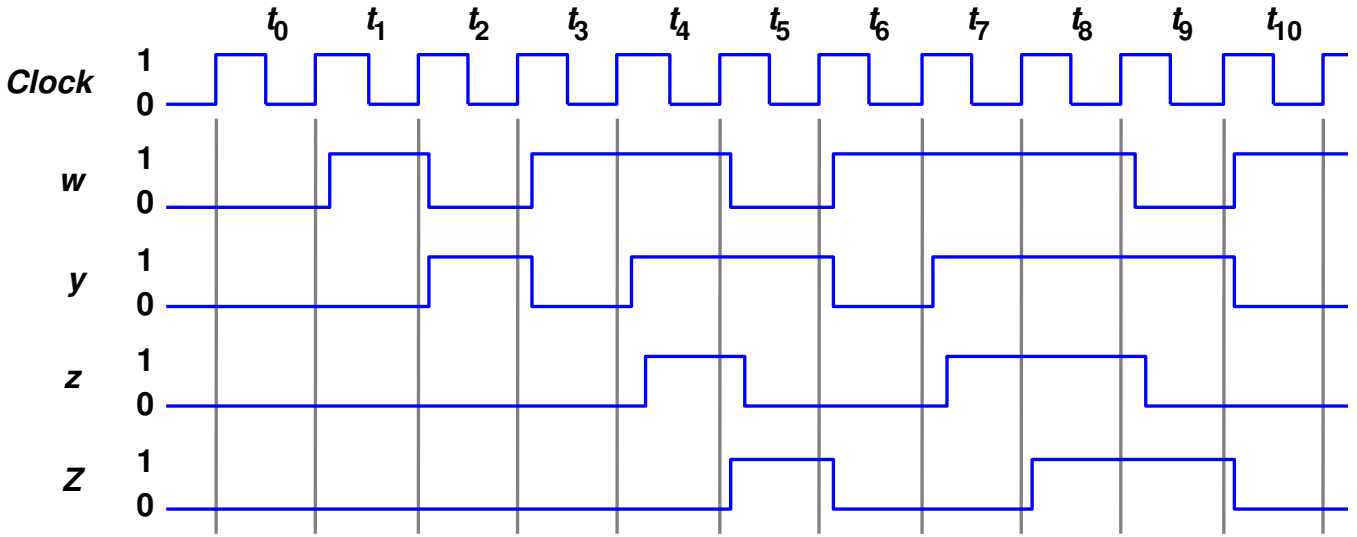
FSM de Mealy


z
w
D
Q
y
Clock
Q̄
Resetn


t0
t1
t2
t3
t4
t5
t6
t7
t8
t9
t10
Clock
1
0
w
1
0
y
1
0
z
1
0

FSM de Mealy
Para a Epecificação Original





FSM - Exercícios

- Projetar um contador binário que conte da seguinte forma: 1, 3, 5, 7, 9, 11, 13, 15, 0, 2, 4, 6, 8, 10 12,14, 1
  - Projete usando FF JK
  - Projete usando FF T
  - Projete usando FF RS

Preenchimento do MK para FF JK e RS

	K		J	
0 -> 0	1 0	X	0 0	0
0 -> 1	0 1	X	1 1	1
1 -> 0	1 1	1	0 1	X
1 -> 1	0 0	0	1 0	X

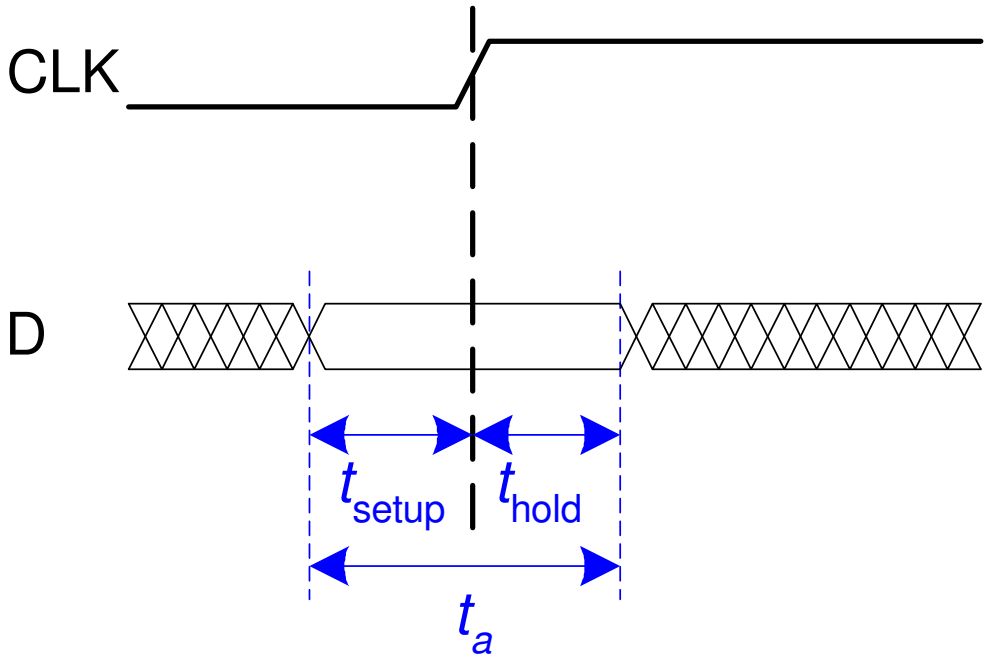
	R		S	
0 -> 0	1 0	X	0 0	0
0 -> 1	0	0	1	1
1 -> 0	1	1	0	0
1 -> 1	0 0	0	1 0	X

Timing em um FF Tipo D

- Flip-flop amostra *D* em cada borda do clock
- *D* deve estar estável quando ele é amostrado
- Similar ao processo fotográfico, *D* deve estar estável em torno da borda do clock
- Se *D* muda quando ele está sendo amostrado pode ocorrer o que chamamos de meta-estabilidade (similar à fotografica ficar "borrada/tremida"

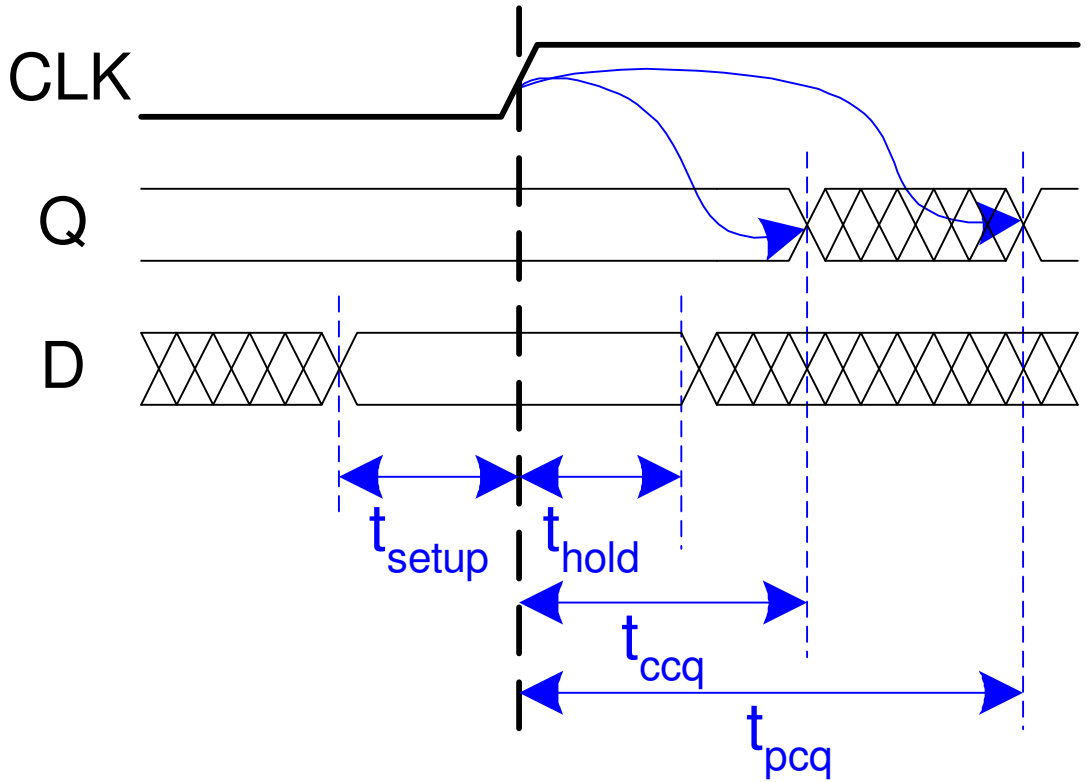
Restrições de Timing de Entrada

- Setup time: t_{setup} = tempo *antes* da borda do clock em que o dado deve permanecer estável (i.e. não mudar)
- Hold time: t_{hold} = tempo *após* a borda do clock em que o dado deve permanecer estável
- Tempo de abertura: t_a = tempo em volta da borda do clock em que o dado deve permanecer estável ($t_a = t_{setup} + t_{hold}$)



Restrições de Timing de Saída

- Propagation delay: t_{pcq} = tempo *após* a borda do clock que é garantido que a saída Q está estável (i.e., para de mudar)
- Contamination delay: t_{ccq} = tempo *após* a borda do clock que Q pode ser instável (i.e., começa a mudar)



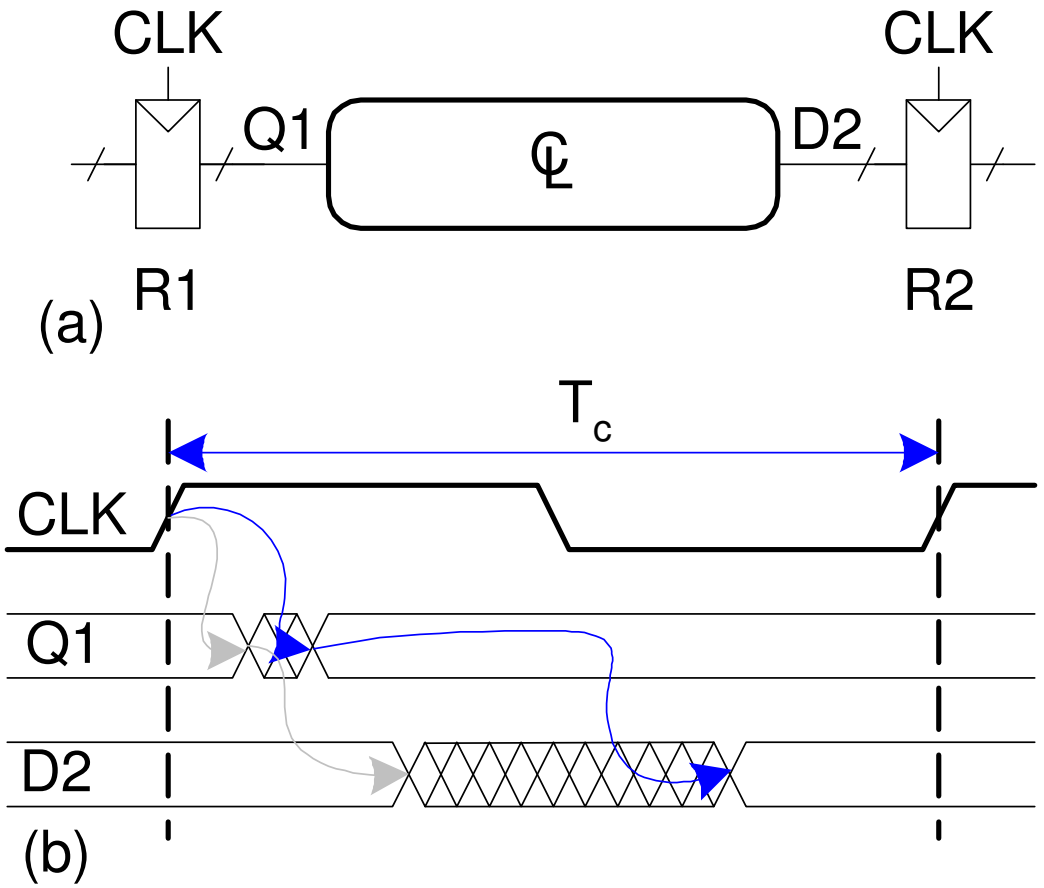
Timing: Comportamento Dinâmico

- A entrada de um circuito seqüencial síncrono deve ser estável durante o tempo de abertura (setup e hold) em volta da borda do clock.

- Especificamente, a entrada deve ser estável:
  - No mínimo t_{setup} *antes* da borda do clock
  - No mínimo t_{hold} *após* a borda do clock

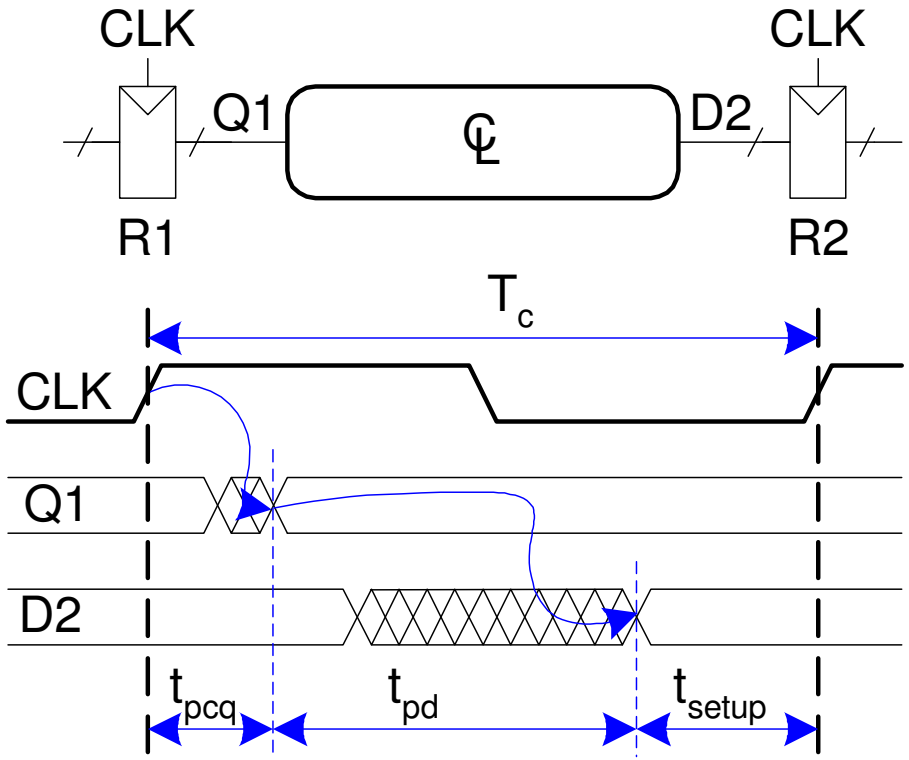
Timing: Comportamento Dinâmico

- O delay entre os registradores tem valores mínimos e máximos, dependendo dos delays dos elementos do circuito



Setup Time

- O setup time depende do delay máximo do registrador R1 e da lógica combinacional.
- A entrada do registrador R2 deve ser estável no mínimo t_{setup} antes da borda do clock.

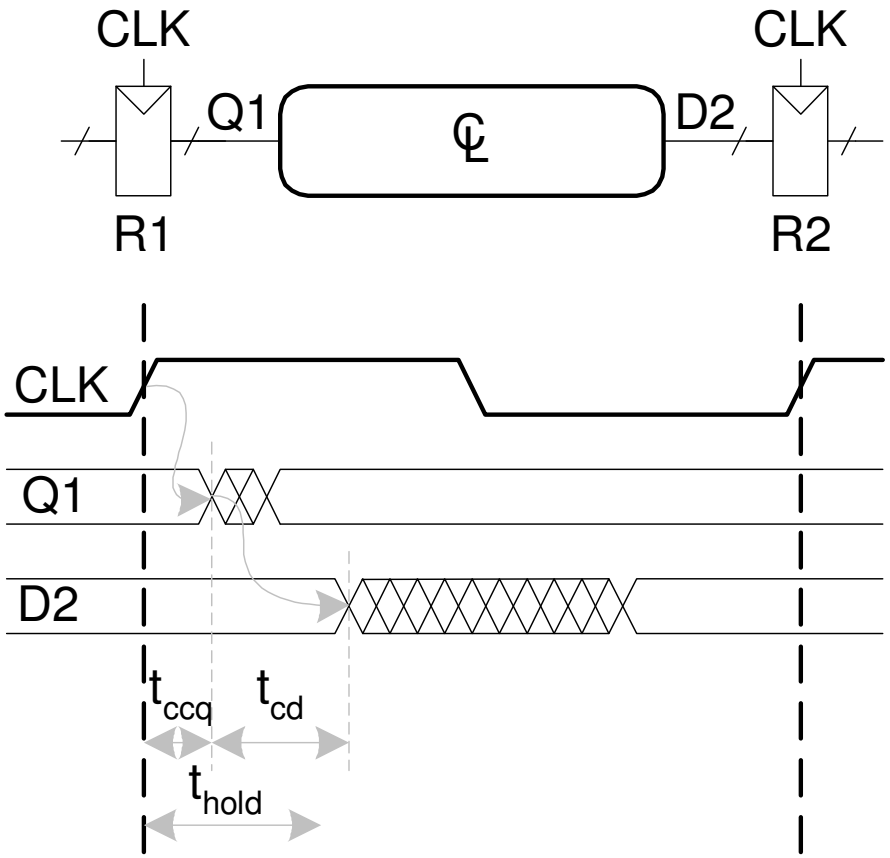


$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

Hold Time

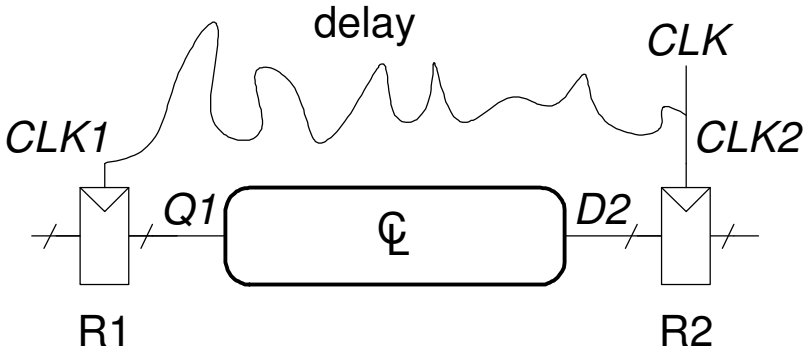
- O hold time depende do delay mínimo do registrador R1 e da lógica combinacional.
- A entrada do registrador R2 deve ser estável por pelo menos t_{hold} após a borda do clock.

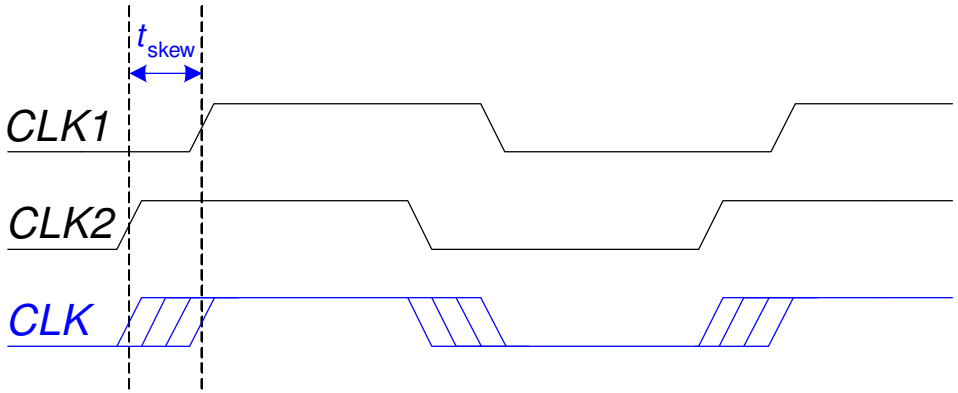


$$t_{ccq} + t_{cd} > t_{hold}$$
$$t_{cd} > t_{hold} - t_{ccq}$$

Clock Skew

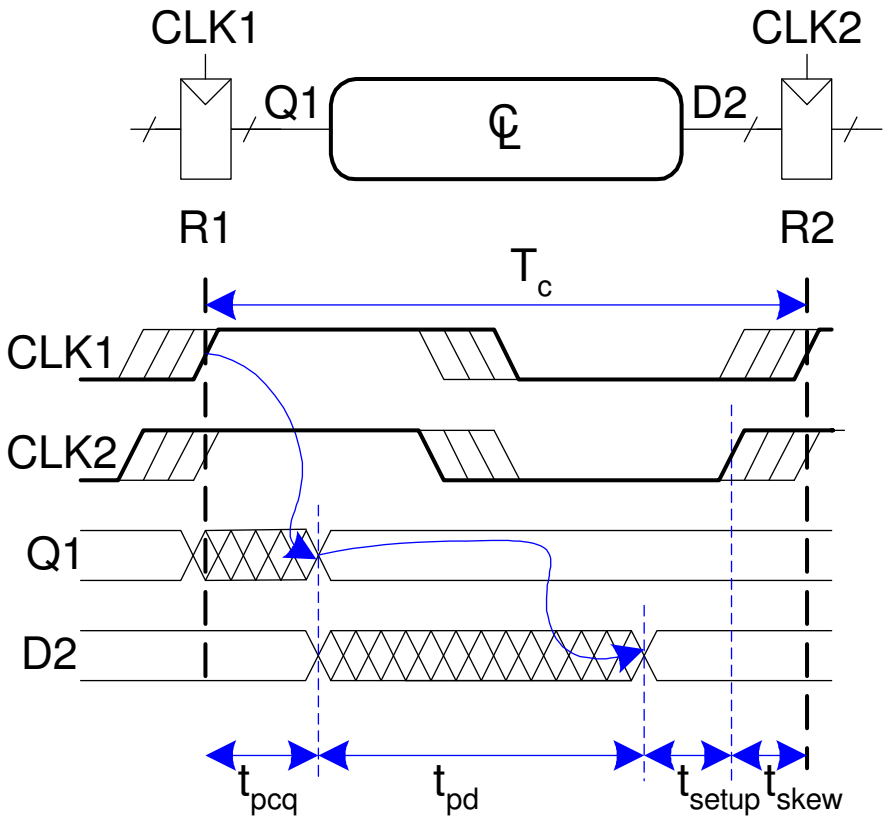
- O clock não chega a todos os registradores ao mesmo tempo
- Isto pode ser causado por atrasos ou ruídos
- Skew é a diferença entre duas bordas de clock
- Quando existem diversos registradores, analisa-se o pior caso e garante-se o seu funcionamento.





Setup Time com Clock Skew

- O pior caso aqui é CLK2 estar adiantado em relação a CLK1

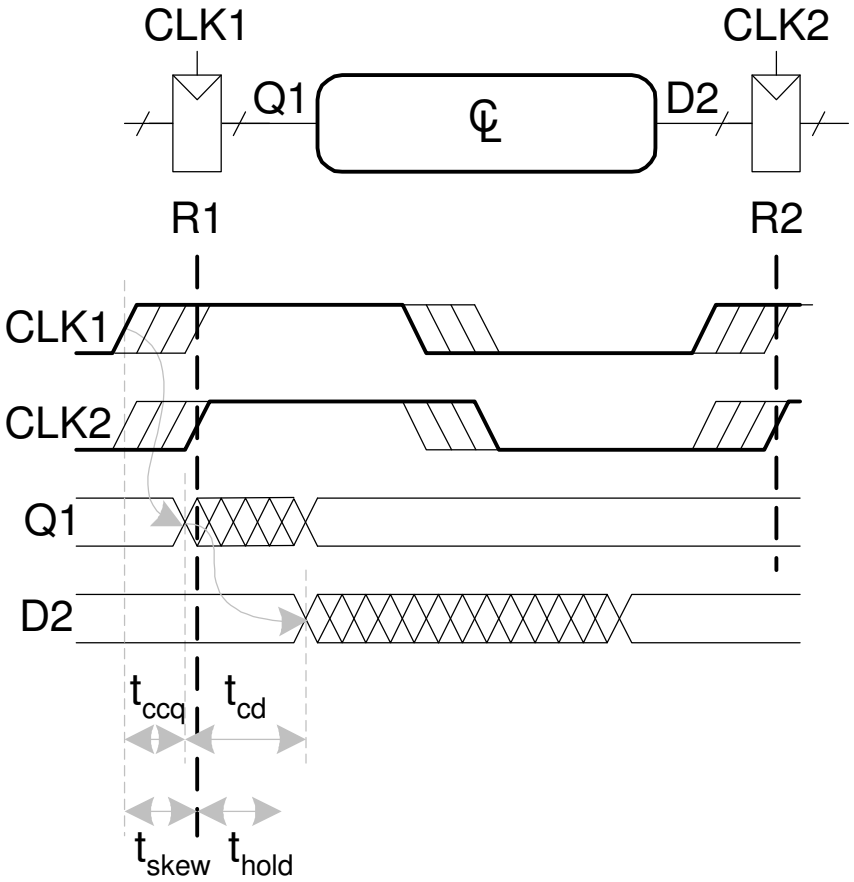


$$T_c - t_{skew} \geq t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

Hold Time com Clock Skew

- O pior caso aqui é CLK2 estar atrasaddo em relação a CLK1



$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$
$$t_{cd} > t_{hold} + t_{skew} - t_{ccq}$$

Clock Skew

- O clock skew tem como efeito aumentar o setup e hold time

- O clock skew reduz o tempo para uso do circuito combinacional

- O clock skew também aumenta o tempo mínimo requerido do circuito combinacional (em geral esse tempo, por decisão de projeto, é zero, o que permite que a saída de um FF possa ser ligada diretamente à entrada de outro FF). Assim, dois FF não podem ser ligados diretamente.

Violando a temporização dinâmica

- **Exemplo: entradas asíncronas podem violar a temporização dinâmica**