

Introducció als Computadors

Tema 6: Circuits Lògics Seqüencials (CLS's)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC6b.pdf>

Enric Morancho
(enricm@ac.upc.edu)

Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

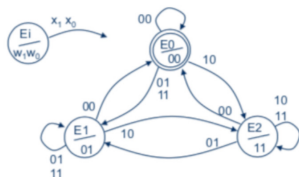
2020-21, 1^{er} quad.

Presentació publicada sota llicència Creative Commons 4.0

Tres formes de representar un CLS



GRAF D'ESTATS



ESQUEMA LÒGIC

		$q_1^* q_0^*$				$w_1 w_0$	
q_1	q_0	$x_1 x_0$	$x_1 x_0$	$x_1 x_0$	$x_1 x_0$		
0	0	0 1	1 0	0 1	0 0	0	0
0	1	0 1	1 0	0 1	0 0	0	1
1	0	1 0	1 0	0 1	0 0	1	1
1	1	x x	x x	x x	x x	x	x

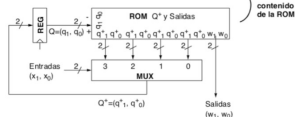


Tabla de transición

$q_1 q_0 x_1 x_0$	$q_1^* q_0^*$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

Tabla de salida

q_1	q_0	w_1	w_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x

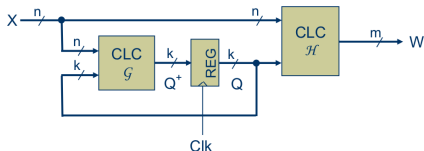
TAULES DE B-RITAT

[1]

- Introducció
- Síntesi en 3 passos de CLS's
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

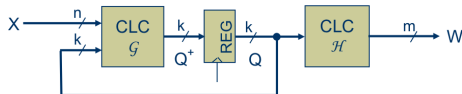
- Circuits Lògics Seqüencials (CLS's) síncrons
 - La sortida depèn de l'entrada i de l'estat emmagatzemat al CLS
 - L'estat s'emmagatzema a uns components denominats biestables
 - Cada biestable pot emmagatzemar un bit
 - Un senyal de rellotge indica els moments en que s'actualitza l'estat
 - Actualització en flanc ascendent d'un senyal periòdic amb període T_c
- Implementarem els CLS's amb 2 CLC's i biestables
 - Un CLC actualitzarà l'estat i l'altre calcularà les sortides
 - A un CLS amb n senyals d'entrada, m de sortida i k bits d'estat
 - CLC que actualitza l'estat: $n + k$ senyals d'entrada i k de sortida
 - CLC que calcula sortida:
Model de Mealy: $n + k$ senyals d'entrada i m de sortida
Model de Moore: k senyals d'entrada i m de sortida
 - k biestables \implies fins a 2^k estats diferents
- Representarem el comportament dels CLS amb 2 TV's o amb un graf
 - Cal saber fer les conversions entre ambdues representacions

Mealy



$$W = \mathcal{H}(Q, X)$$

Moore



$$W = \mathcal{H}(Q)$$

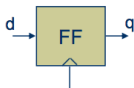
- Els dos models tenen la mateixa potència
- Diferències:
 - Moore pot necessitar més estats que Mealy per fer la mateixa tasca
 - Moore triga, com a mínim, un cicle en propagar a la sortida un canvi a l'entrada

- Introducció
- Síntesi en 3 passos de CLS's
 - De la descripció funcional al graf d'estats
 - Del graf d'estats a les taules de transicions i sortides
 - De les taules a l'esquema lògic
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

- Introducció
- Síntesi en 3 passos de CLS's
 - De la descripció funcional al graf d'estats
 - Del graf d'estats a les taules de transicions i sortides
 - De les taules a l'esquema lògic
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

- Donada la descripció funcional del CLS com a text, pseudocodi, expressions algebraiques,... determinar el graf d'estats del CLS
 - També ens poden aportar un cronograma simplificat que mostri el comportament desitjat a alguns casos
- No existeix un mètode sistemàtic
 - És el pas més difícil a la síntesi del CLS
- La solució pot no ser única

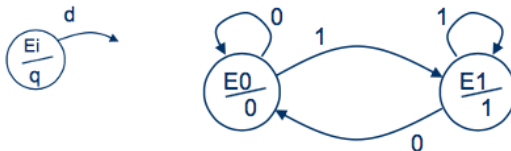
- Determinar el graf d'estats del CLS més senzill, el biestable D.
- L'encapsulament i un cronograma simplificat d'exemple seria:
 - $n = m = 1$



Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Entrada d	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0
Salida q	0	1	1	0	1	0	0	0	1	0	1	0	0	0	1

- Graf d'estats resultant:
 - Dos estats perquè el biestable pot emmagatzemar els valors "0" i "1"

Leyenda:




- En concordància amb el cronograma, l'estat inicial seria E0

Exemple 2: reconeixedor de 011 (CLS-011)



- Dibuixeu el graf d'estats d'un CLS que reconegui la seqüència 011 com mostra el cronograma simplificat.



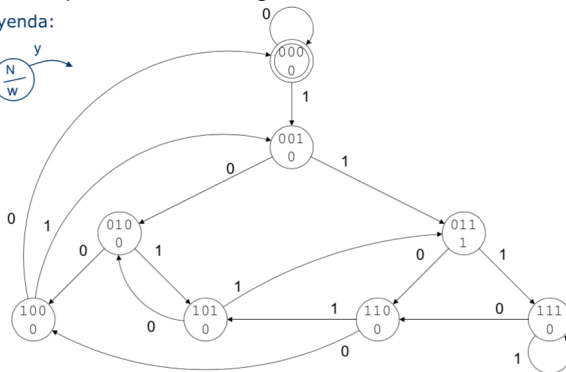
Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Entrada y	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0
Salida w	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0

- De l'encapsulament deduïm $n = m = 1$

- L'estat del CLS-011 serà el valor de l'entrada els tres darrers cicles

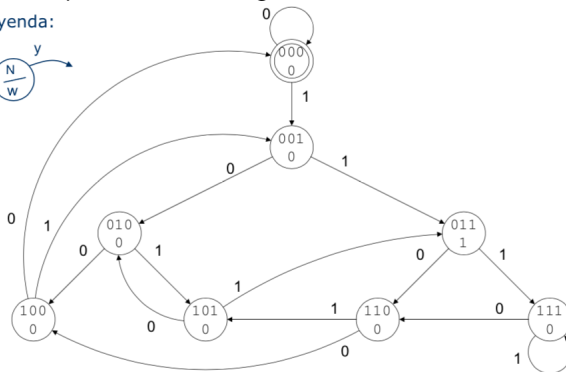
- $N = (y(t-3)y(t-2)y(t-1))$
- 8 estats possibles ($k = \lceil \log_2 8 \rceil = 3$ bits d'estat)
 - {"000", "001", "010", "011", "100", "101", "110", "111"}
 - Decidim que l'estat inicial sigui "000"

Leyenda:



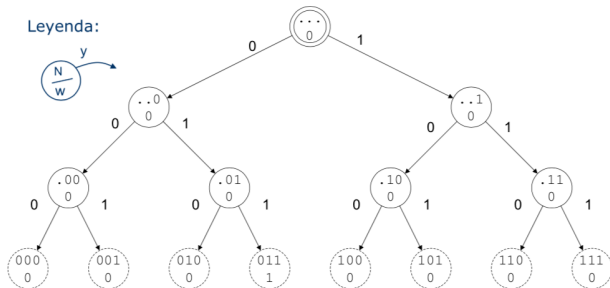
- L'estat del CLS-011 serà el valor de l'entrada els tres darrers cicles
 - $N = (y(t-3)y(t-2)y(t-1))$
 - 8 estats possibles ($k = \lceil \log_2 8 \rceil = 3$ bits d'estat)
 - $\{ "000", "001", "010", "011", "100", "101", "110", "111" \}$
 - Decidim que l'estat inicial sigui "000"

Leyenda:



- I si l'entrada comença per "11"?
 - Tenim un "fals positiu" (*false positive*)

- L'estat del CLS-011 serà el valor de l'entrada els tres darrers cicles, tenint en compte que als primers tres cicles no disposem d'aquesta informació
 - 15 estats possibles ($k = \lceil \log_2 15 \rceil = 4$ bits d'estat)
 - $\{ "000", "001", "010", "011", "100", "101", "110", "111" \}$
 - $\{ "...", "..0", "..1", ".00", ".01", ".10", ".11" \}$
 - L'estat inicial serà $"..."$
- Fragment del graf (cal completar-lo amb el graf anterior)

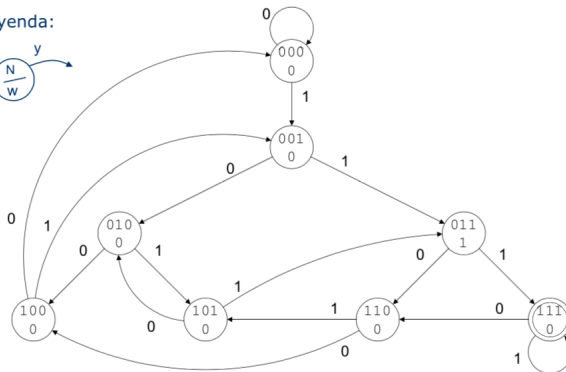


Exemple 2 CLS-011: tercera aproximació



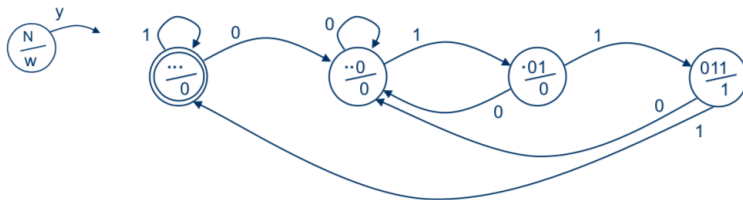
- Tornem a la primera aproximació del CLS-011 modificant l'estat inicial
 - Decidim que l'estat inicial sigui "111" en comptes de "000"
 - Fa que desaparegui el fals positiu causat per l'entrada "11"

Leyenda:



- Intentarem reduir el nombre d'estats del CLS-011 a 4
 - L'estat indicarà la porció de "011" reconeguda els tres darrers cicles
 - "... " (res), "..0" (el darrer cicle hem rebut "0"), ".01" (els dos darrers cicles hem rebut "01") i "011" (seqüència detectada)
 - L'estat inicial serà "..."

Leyenda:



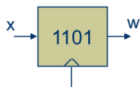
- Es pot demostrar que 4 estats és l'òptim per a aquest problema
 - $k = \lceil \log_2 4 \rceil = 2$ bits de estat
 - La implementació només necessitarà dos biestables

- Dibuixeu el graf d'estats d'un CLS que reconegui la seqüència 1101 com mostra el cronograma simplificat.

- Considerarem dues variants:

- Sense solapament
- Amb solapament

a) Sin solapamiento



Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Entrada y	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1
Salida w	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

b) Con solapamiento

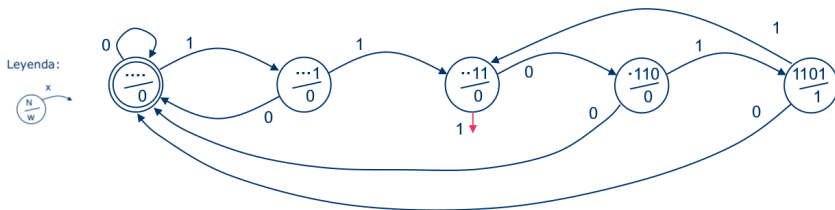
Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Entrada y	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1
Salida w	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1

- De l'encapsulament deduïm $n = m = 1$
- Al cas "amb solapament", l'"1" del cycle 6 pertany a dues seqüències

- b) Amb solapament**

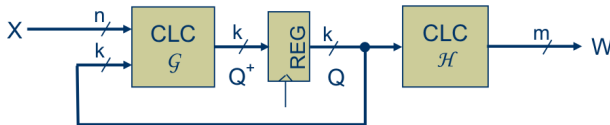


- Si alguna combinació d'entrades és impossible quan el CLS està a algun estat, ho indicarem amb un arc que no vagi a parar a cap node
 - Exemple: el reconeixedor de "1101" amb solapament però ara ens garanteixen que a l'entrada mai apareixeran tres "1" seguits
 - Si estem a l'estat "..11", no podrà arribar un nou "1"

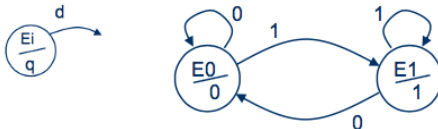


- Introducció
- Síntesi en 3 passos de CLS's
 - De la descripció funcional al graf d'estats
 - Del graf d'estats a les taules de transicions i sortides
 - De les taules a l'esquema lògic
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

- Ja hem vist com fer el pas invers
 - Nodes del graf representen estats
 - Un arc representa el canvi d'estat que provoca un valor de l'entrades
- Nombre de biestables? (k)
 - $k = \lceil \log_2 \text{nombre_estats} \rceil$
 - Amb 13 estats $\implies k = \lceil \log_2 13 \rceil = 4$
- Mida de les taules (model de Moore):
 - \mathcal{G} (transicions): $n + k$ bits d'entrada, k bits de sortida
 - Taula de veritat de 2^{n+k} files $\times k$ columnes
 - \mathcal{H} (sortides): k bits d'entrada, m bits de sortida
 - Taula de veritat de 2^k files $\times m$ columnes

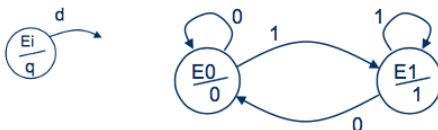


Llegenda:



- Per la llegenda ja sabem que $n = m = 1$
- Com tenim dos estats $\implies k = \lceil \log_2 2 \rceil = 1$
- Cal codificar cada estat amb un vector de k bits
 - Identificador d'estat
 - Si no ens la imposen, tenim plena llibertat per fer la codificació
 - Algunes codificacions simplificaran la generació de les sortides
 - Codificació triada:
 - $E0 = "0"$
 - $E1 = "1"$

Leyenda:



- $n = m = k = 1$
 - E0 és $Q = 0$, E1 és $Q = 1$
- Construïm les taules
 - Per conveni, l'identificador d'estat té més pes que les entrades del CLS

Transicions:

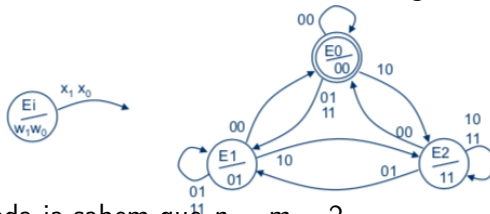
Q	d	Q^+
0	0	0
0	1	1
1	0	0
1	1	1

Sortides:

Q	q
0	0
1	1

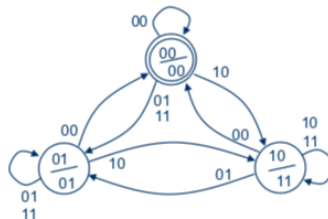
- Ens adonem que l'estat futur és independent de l'estat actual
- Amb aquesta codificació d'estats, la funció de sortida és trivial

- Obtenir les taules de transicions i sortides del graf:



- Per la llegenda ja sabem que $n = m = 2$
- Com tenim tres estats $\implies k = \lceil \log_2 3 \rceil = 2$
- Un cop codificats els estats amb k bits el graf queda:

Ei	$q_1 q_0$
E0	00
E1	01
E2	10



Exemple 2: obtenir taules

Ei	$q_1 q_0$
E0	00
E1	01
E2	10



Tabla de transición

$q_1 q_0 x_1 x_0$	$q_1^* q_0^*$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

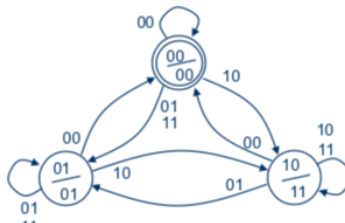


Tabla de salida

q_1	q_0	w_1	w_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x

- Introducció
- Síntesi en 3 passos de CLS's
 - De la descripció funcional al graf d'estats
 - Del graf d'estats a les taules de transicions i sortides
 - De les taules a l'esquema lògic
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

- Ja sabem com sintetitzar un CLC
 - Suma de *minterms*
 - Decodificador
 - ROM
- Ho aplicarem a cada CLC del CLS
- També veurem com sintetitzar els dos CLC's amb una única ROM i multiplexors de busos

- Obtenir l'esquema lògic corresponent a un CLS amb aquestes taules de transicions i sortides ($n = m = k = 2$)
 - Sintetitzarem un CLC amb una ROM i l'altre amb un descodificador

Tabla de transición

$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

Tabla de salida

q_1	q_0	w_1	w_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x

\Rightarrow

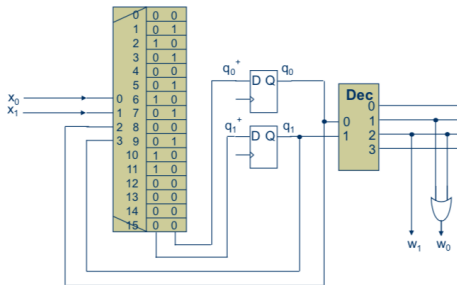
- Obtenir l'esquema lògic corresponent a un CLS amb aquestes taules de transicions i sortides ($n = m = k = 2$)
 - Sintetitzarem un CLC amb una ROM i l'altre amb un descodificador

Tabla de transición

$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

Tabla de salida

q_1	q_0	w_1	w_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x



- Sintetitzarem els dos CLC's amb una única ROM
 - Reorganitzem el contingut de les dues TV's en una de sola

Tabla de transición

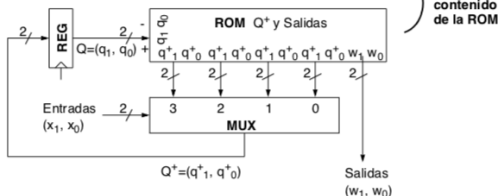
$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

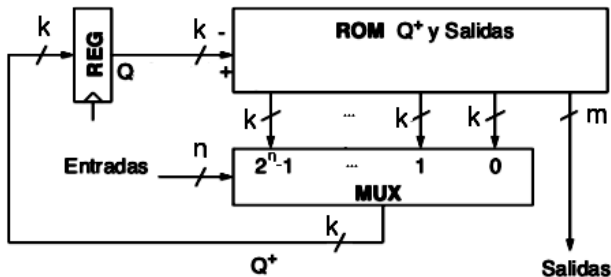
Tabla de salida

$q_1 q_0$	$w_1 w_0$
0 0	0 0
0 1	0 1
1 0	1 1
1 1	x x



$q_1 q_0$	$q_1^+ q_0^+$				$w_1 w_0$	
	$x_1 x_0$ 1 1	$x_1 x_0$ 1 0	$x_1 x_0$ 0 1	$x_1 x_0$ 0 0		
0 0	0 1	1 0	0 1	0 0	0 0	
0 1	0 1	1 0	0 1	0 0	0 1	
1 0	1 0	1 0	0 1	0 0	1 1	
1 1	x x	x x	x x	x x	x x	

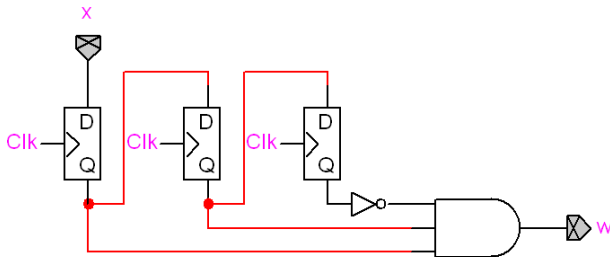




- La quantitat d'informació és la mateixa tant si es sintetitza amb una ROM com si es fa amb dues
 - CLC's sintetitzats amb dues ROMs
 - $2^{k+n} \times k + 2^k \times m$ bits
 - CLC's sintetitzats amb una única ROM
 - $2^k \times (2^n \times k + m) = 2^{k+n} \times k + 2^k \times m$ bits

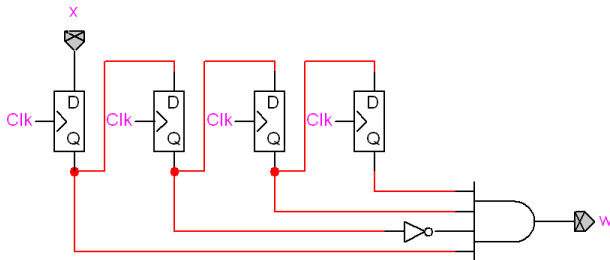
- Introducció
- Síntesi en 3 passos de CLS's
- **Síntesi ad-hoc de CLS's**
- Exercicis
- Conclusions

- Implementació amb *shift register* de tres bits
 - Assumim biestables inicialitzats a "1" per a evitar el fals positiu



- És el que obtindríem si sintetitzem (i minimitzem amb Karnaugh) el graf amb 8 estats proposat
- No és la solució òptima, hem vist que es podia fer amb 2 biestables

- Implementació amb *shift register* de quatre bits



- Reconeix la seqüència 1101 amb solapament o sense solapament?
- Per a l'altre cas, sintetitzeu un CLS basat també en un *shift register*

- Introducció
- Síntesi en 3 passos de CLS's
- Síntesi ad-hoc de CLS's
- Exercicis
- Conclusions

- Es vol dissenyar un CLS que permeti gestionar un dau electrònic
 - L'entrada del CLS (*jugar*) serà generada per un pulsador
 - Les sortides estaran connectades a un component electrònic amb 7 entrades (una per a cada punt del dau); cada entrada regula un *led*

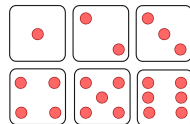


[2]

x_6		x_5
x_4	x_3	x_2
x_1		x_0

ON $\iff x_i = "1"$

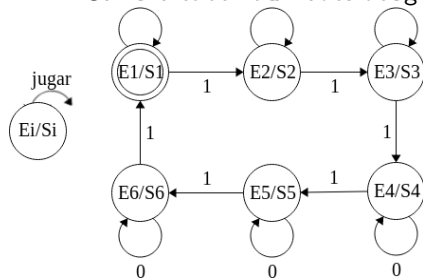
OFF $\iff x_i = "0"$



[3]

- Descripció:
 - Mentre el senyal d'entrada *jugar* valgui "1", el CLS mostrarà cíclicament els 6 resultats possibles (començant per l'1) a raó d'un per cycle
 - Mentre *jugar* valgui 0, la combinació mostrada al *display* no variarà
- Dibuixeu el graf d'estats del CLS
 - Inicialment, es mostrarà el valor 1

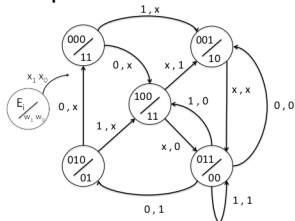
- Interfície del CLS: entrada *jugar* d'1 bit, sortida *W* de 7 bits
- Nombre d'estats: 6 (un estat per a cada valor a visualitzar al dau)
 - $\{E1, E2, E3, E4, E5, E6\}$, $E1$ és l'estat inicial
- Transicions entre estats:
 - Si *jugar*="1", canviarà d'estat cíclicament
 - Si *jugar*="0", no canviarà d'estat
- Sortides:
 - A cada estat, activar els *leds* necessaris per a visualitzar el nombre
 - Comono caben als nodes del graf, les posarem a taula adjacent



	w_6	w_5	w_4	w_3	w_2	w_1	w_0
S1	0	0	0	1	0	0	0
S2	1	0	0	0	0	0	1
S3	1	0	0	1	0	0	1
S4	1	1	0	0	0	1	1
S5	1	1	0	1	0	1	1
S6	1	1	1	0	1	1	1

- Variants sobre el graf anterior (són independents):
 - a) Modifiquen el graf d'estats de forma que el *display* mostri el valor únicament quan *jugar* valgui "0"; mentre valgui "1", el CLS també actualitzarà cíclicament el valor però no es mostrarà al *display*.
 - b) Modifiquen el graf d'estats de forma que quan *jugar* passi a valdre "1", si el valor actual del comptador és parell, el CLS passarà a generar la seqüència de valors cíclicament en ordre descendent (6,5,...,1,6,...) a partir del valor actual. Si el valor actual és senar, passarà a generar la seqüència cíclicament en ordre ascendent (1,2,...,6,1,...) a partir del valor actual.
 - c) Modifiquen el graf d'estats de forma que el CLS tingui una segona entrada anomenada *sentit*. Si val "0", el CLS generarà la seqüència de valors en ordre ascendent i si val "1" la generarà en ordre descendent; el canvi de *sentit* s'aplicarà a partir del valor actual.
 - d) Modifiquen el graf d'estat de forma que el CLS garanteixi que quan *jugar* val "0" el valor actual es mostri, com a mínim, tres cicles consecutius. Un cop transcorreguts, en el moment que *jugar* passi a valdre "1", el CLS tornarà a generar la seqüència de valors.

- Ompliu la taula de sortides i un fragment de la de transicions



q2	q1	q0	x1	x0	q2 ⁺	q1 ⁺	q0 ⁺
0	1	1	0	0			
0	1	1	0	1			
0	1	1	1	0			
0	1	1	1	1			
1	0	0	0	0			
1	0	0	0	1			
1	0	0	1	0			
1	0	0	1	1			

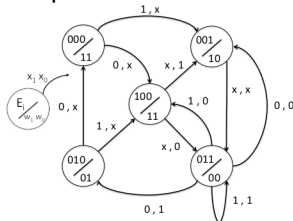
q2	q1	q0	w1	w0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- Ompliu el següent cronograma simplificat

Cicle	0	1	2	3	4	5	6	7	8
x₁ x₀	00	10	01	11	00	10	01	00	11
q₂ q₁ q₀	000								
w₁ w₀									

- El sintetitzem amb una ROM i un multiplexor de bussos, quants biestables calen?, quantes paraules tindrà la ROM?, quants bits per paraula?, quantes entrades tindrà el multiplexor?, quants bits per bus?

- Ompliu la taula de sortides i un fragment de la de transicions



q2	q1	q0	x1	x0	q2 ⁺	q1 ⁺	q0 ⁺	q2	q1	q0	w1	w0
0	1	1	0	0	0	0	1	0	0	0	1	1
0	1	1	0	1	0	1	0	0	0	1	1	0
0	1	1	1	0	1	0	0	0	1	0	0	1
0	1	1	1	1	0	1	1	0	1	1	0	0
1	0	0	0	0	0	1	1	1	0	0	1	1
1	0	0	0	1	0	0	1	1	0	1	x	x
1	0	0	1	0	0	1	1	1	1	0	x	x
1	0	0	1	1	0	0	1	1	1	1	x	x

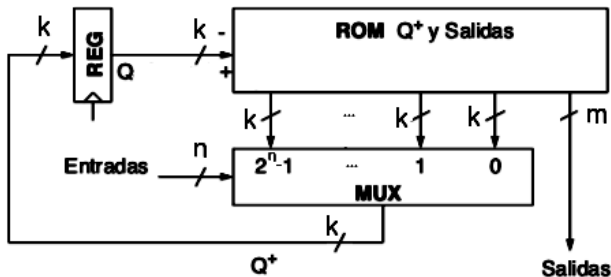
- Ompliu el següent cronograma simplificat

Cicle	0	1	2	3	4	5	6	7	8
x₁ x₀	00	10	01	11	00	10	01	00	11
q₂ q₁ q₀	000	100	011	010	100	011	100	001	011
w₁ w₀	11	11	00	01	11	00	11	10	00

- El sintetitzem amb una ROM i un multiplexor de bussos, quants biestables calen?, quantes paraules tindrà la ROM?, quants bits per paraula?, quantes entrades tindrà el multiplexor?, quants bits per bus
 - 3 biestables, 8 paraules, 14 bits/paraula, 4 entrades, 3 bits per bus

- Introducció
- Síntesi en 3 passos de CLS's
- Síntesi ad-hoc de CLS's
- Exercicis
- **Conclusions**

- Hem vist la síntesi de CLS's en tres passos:
 - 1 Construir el graf d'estats a partir de la descripció funcional
 - 2 Definir les TV's dels CLC's a partir del graf d'estats
 - 3 Sintetitzar el CLS a partir de les TV's dels CLC's
- Tenim la possibilitat de sintetitzar els dos CLC's d'un CLS utilitzant una única ROM i un multiplexor de busos



- No oblideu fer l'ET6b a Atenea abans de la propera classe

Llevat que s'indiqui el contrari, les figures, esquemes, cronogrames i altre material gràfic o bé han estat extrets de la documentació de l'assignatura elaborada per Juanjo Navarro i Toni Juan, o corresponen a enunciats de problemes i exàmens de l'assignatura, o bé són d'elaboració pròpia.

- [1] [Online]. Available:
<http://cherryblossom-books.com/index.php/godel-escher-bach-an-eternal-golden-braid.html>.
- [2] [Online]. Available: <https://www.sparkfun.com/products/10212>.
- [3] [Online]. Available: <https://commons.wikimedia.org/wiki/File:Dice.png>.

Introducció als Computadors

Tema 6: Circuits Lògics Seqüencials (CLS's)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC6b.pdf>

Enric Morancho
(enricm@ac.upc.edu)

Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

2020-21, 1^{er} quad.

Presentació publicada sota llicència Creative Commons 4.0