

# Introducció als Computadors

## Tema 6: Circuits Lògics Seqüencials (CLS's)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC6a.pdf>

Enric Morancho  
([enricm@ac.upc.edu](mailto: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<sup>er</sup> quad.

Presentació publicada sota llicència Creative Commons 4.0

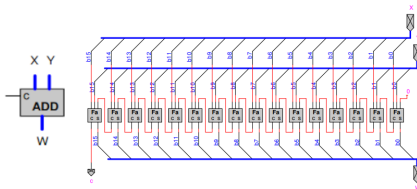


[1]

- Introducció
  - Concepte de Circuit Lògic Seqüencial (CLS)
  - Components per a l'emmagatzemament d'informació
  - Regles d'interconnexió als CLS's
  - Temps de cicle d'un CLS
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Introducció
  - Concepte de Circuit Lògic Seqüencial (CLS)
    - Components per a l'emmagatzemament d'informació
    - Regles d'interconnexió als CLS's
    - Temps de cicle d'un CLS
  - Estructura general d'un CLS
  - Graf d'estats
  - Exemples de CLS's
  - Conclusions
  - Autoaprenentatge

- Als CLC's, les sortides només depenen del valor actual de les entrades
  - La història prèvia no influeix en les sortides
    - Per a algunes tasques, ja ens està bé. Per exemple: bloc sumador
    - Però per a d'altres tasques pot ser necessari mantenir història.  
Exemple: el controlador d'un "su turno"



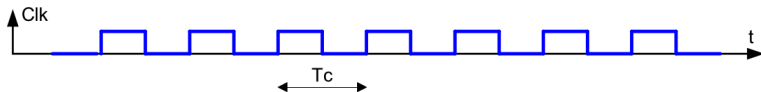
[2]

- Als CLS's (Circuits Lògics Seqüencials) les sortides depenen del valor actual de les entrades i de la informació emmagatzemada al CLS

- Veurem que tot CLS estarà format per:
  - Biestables: components que permeten emmagatzemar informació
    - Cada biestable permet emmagatzemar 1 bit
    - Estat (*state*) del CLS: informació emmagatzemada als biestables
    - $k$  biestables  $\implies k$  bits d'estat  $\implies$  fins a  $2^k$  estats diferents
  - Lògica combinacional (CLC)
    - Calcula les sortides i el nou estat a partir de les entrades i de l'estat actual
- En termes de funcions lògiques:
  - CLC:
    - $output = f(input)$
  - CLS:
    - $output = f_1(input, state)$
    - $state = f_2(input, state)$
- El computador que construïrem al llarg del curs serà un CLS

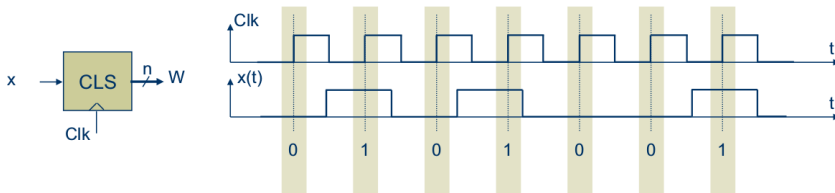
- CLS's Síncrons (*clock driven*)

- Utilitzen un senyal de control, senyal de rellotge (*clock, clk*)
  - Senyal periòdica de període  $T_c$  unitats de temps
  - El període té un subperíode on el senyal val "1" i un altre on val "0"



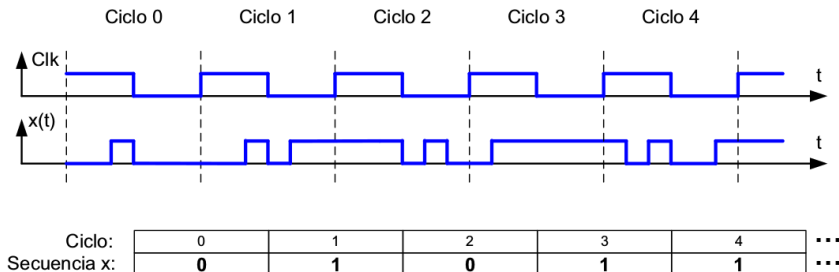
- La freqüència del senyal és  $Freq = 1/T_c$  cicles per unitat de temps
    - A un computador on  $Freq = 1GHz$ ,  $T_c = 1/Freq = 10^{-9}s = 1ns$
  - L'estat només es podrà actualitzar cada  $T_c$  unitats de temps
  - El  $T_c$  haurà de ser més gran que el  $T_p$  del CLC que calcula el nou estat
  - Són els que majoritàriament utilitzarem a IC
- CLS's Asíncrons (*event driven*)
    - L'estat es pot modificar en qualsevol moment
    - Només els utilitzarem al tema d'entrada/sortida

- A un CLS síncron tidrem un únic senyal de rellotge
- Quan es produeixi un flanc ascendent en el senyal (passi de "0" a "1") assumirem que els CLC's han completat el seu càlcul i que les entrades dels dispositius d'emmagatzemament tenen un valor correcte i estable
  - És fiable emmagatzemar el seu valor als dispositius d'emmagatzematge
- El senyal de rellotge ens servirà de referència per a interpretar els senyals d'entrada
  - Tindrem en compte el valor del senyal  $x$  en el moment que es produeixin flancs ascendents del senyal de rellotge



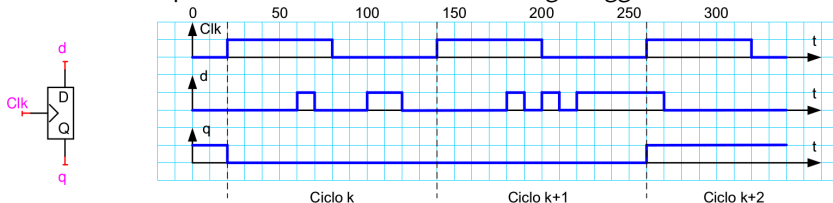


- Com el valor del senyal només és significatiu en el moment que es produeix un flanc ascendent de rellotge, simplifiquem el cronograma i el reduïm als instants en els que es produeixen flancs ascendents
  - Una altra discretització del senyal
  - Cronograma simplificat

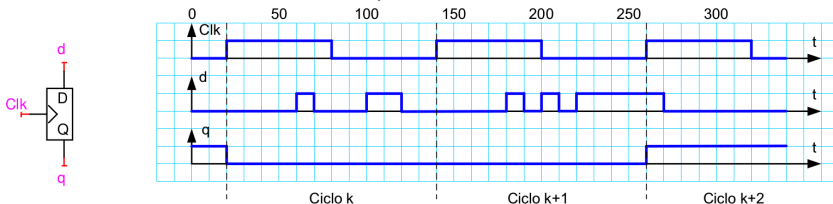


- Introducció
  - Concepte de Circuit Lògic Seqüencial (CLS)
  - Components per a l'emmagatzemament d'informació
    - Regles d'interconnexió als CLS's
    - Temps de cicle d'un CLS
  - Estructura general d'un CLS
  - Graf d'estats
  - Exemples de CLS's
  - Conclusions
  - Autoaprenentatge

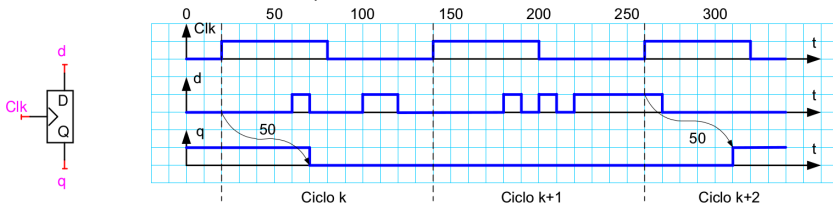
- És el component bàsic dels CLS's
- Dels biestables existents, a IC considerarem el biestable D
  - També conegut com a *Data / Delay / flip-flop*
  - 2 senyals d'entrada (D i rellotge (*clk*)), i un senyal de sortida (Q)
  - Pot tenir definit un valor inicial
- Quan es produeix un flanc ascendent del senyal de rellotge, el valor actual de D passa a Q i **es manté estable fins al proper flanc ascendent de rellotge**, moment en que es repeteix el procés.
  - Activat per flanc ascendent = *Positive edge triggered*



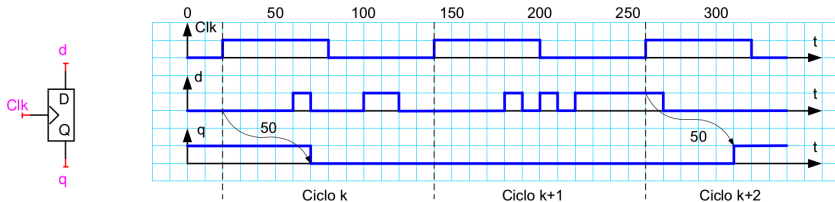
- Com les portes lògiques, els biestables tenen un temps de propagació
  - Comportament idealitzat ( $T_p = 0 \text{ u.t.}$ )



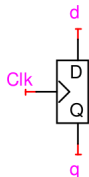
- Comportament amb  $T_p = 50 \text{ u.t.}$



- Garanteix un valor estable a la sortida  $q$  al llarg d'un cicle
- Filtra els *glitches* que pugui mostrar el senyal  $d$ , el senyal  $q$  és estable

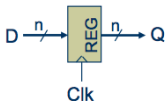


- Retarda el senyal d'entrada  $d$  un cicle
  - Exemple: cronograma simplificat del biestable D amb entrada 001



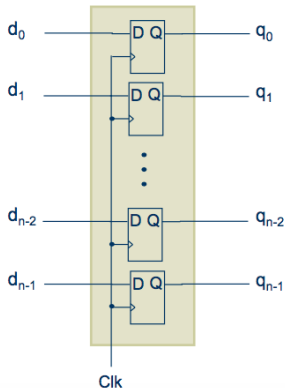
Ciclo:	...	K-1	k	K+1	K+2	...
d:	...	0	0	1		...
q:	...		0	0	1	...

- Un registre de  $n$  bits (REG- $n$ ) és un CLS format per  $n$  biestables que comparteixen el senyal de rellotge
  - Tots  $n$  biestables s'actualitzaran alhora



$D = d_{n-1}, d_{n-2}, \dots, d_1, d_0$

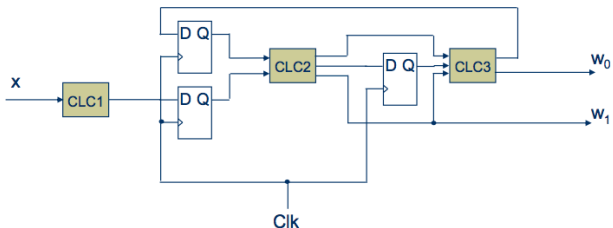
$Q = q_{n-1}, q_{n-2}, \dots, q_1, q_0$



## ● Introducció

- Concepte de Circuit Lògic Seqüencial (CLS)
- Components per a l'emmagatzemament d'informació
- Regles d'interconnexió als CLS's
  - Temps de cicle d'un CLS
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Varem establir tres regles d'interconnexió de CLC's
  - No existència de cicles
  - Totes les entrades han d'estar definides
  - No podem curtcircuitar sortides
- Als CLS's n'introduïm alguns canvis
  - Podrem tenir cicles
    - Sempre que el cicle travessi, com a mínim, un biestable o un registre



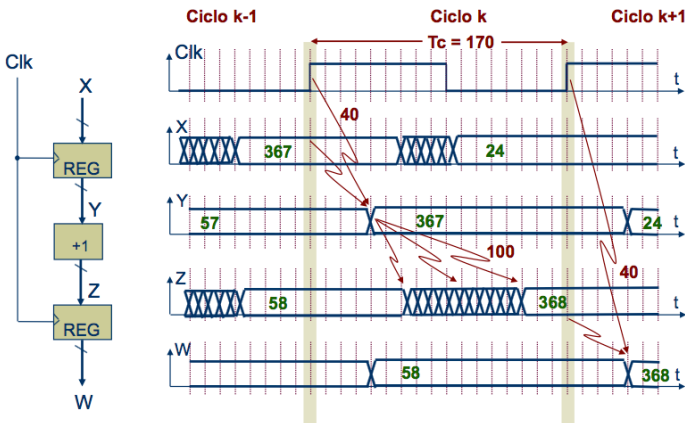
- El senyal *clk* s'ha de connectar a tots els registres i biestables
  - Normalment no el dibuixarem explícitament



## ● Introducció

- Concepte de Circuit Lògic Seqüencial (CLS)
- Components per a l'emmagatzemament d'informació
- Regles d'interconnexió als CLS's
- Temps de cicle d'un CLS
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- $T_p(REG) = 40 \text{ u.t.}$ ,  $T_p(+1) = 100 \text{ u.t.}$ ,  $T_{cicle} = 170 \text{ u.t.}$
- El registre ens garanteix que l'entrada del bloc +1 és **estable**
- El temps de cicle ( $T_c$ ) ha de permetre propagar el senyal des de la sortida d'un registre fins a l'entrada del següent registre
  - $T_c \geq T_p(REG) + T_p(+1)$



- Introducció
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Separarem la lògica combinacional dels biestables
  - La lògica combinacional estarà encapsulada dins d'un o de dos CLC's
- Veurem dos models de CLS's:
  - Mealy
    - Cas general
  - Moore
    - Cas particular del model de Mealy
    - És el que farem servir a l'assignatura

George H. Mealy (1927-2010)

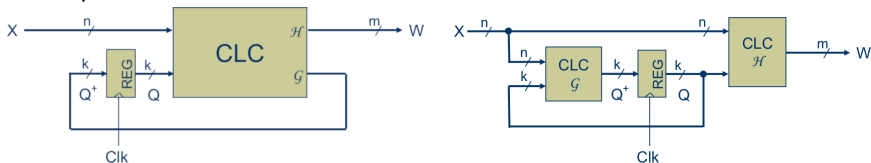
Edward F. Moore (1925-2003)

?



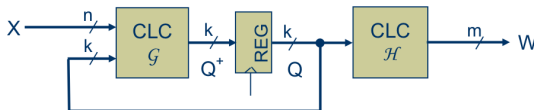
[3]

- Es pot sintetitzar amb 1 CLC o amb 2 CLC's



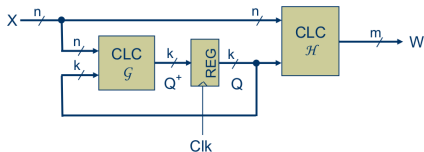
- Especificació del CLS:
  - $n$  bits de entrada,  $m$  bits de sortida,  $k$  bits d'estat
    - $Q$ : estat actual,  $Q^+$ : estat futur
    - Pot representar fins a  $2^k$  estats diferents
    - Cal determinar l'estat inicial**
- Especificació del CLC:
  - Esquema amb 1 CLC:
    - $n + k$  bits d'entrada,  $m + k$  bits de sortida
  - Esquema amb 2 CLCs:
    - CLC  $\mathcal{G}$  (transicions):  $n + k$  bits d'entrada,  $k$  bits de sortida
    - CLC  $\mathcal{H}$  (sortides):  $n + k$  bits d'entrada,  $m$  bits de sortida

- És el que utilitzarem a l'assignatura
  - La sortida depèn únicament de l'estat actual



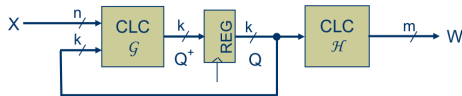
- Especificació del CLS:
  - $n$  bits de entrada,  $m$  bits de sortida,  $k$  bits d'estat
    - $Q$ : estat actual,  $Q^+$ : estat futur
    - Pot representar fins a  $2^k$  estats diferents
    - **Cal determinar l'estat inicial**
- Especificació dels CLC's:
  - CLC  $\mathcal{G}$  (transicions):  $n + k$  bits d'entrada,  $k$  bits de sortida
    - Taula de veritat de  $2^{n+k}$  files  $\times k$  columnes
  - CLC  $\mathcal{H}$  (sortides):  $k$  bits d'entrada,  $m$  bits de sortida
    - Taula de veritat de  $2^k$  files  $\times m$  columnes

## Mealy



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

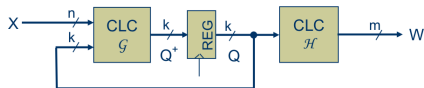
## Moore



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

- 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

- $n = m = k = 2$ , l'estat inicial és  $q_1 q_0 = 00$
- Degut el contingut de la taula  $Q^+$ , mai arribarem a l'estat  $q_1 q_0 = 11$ 
  - Estat inassolible (*unreachable*)  $Q^+ = \mathcal{G}(Q, X)$ :



$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
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

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

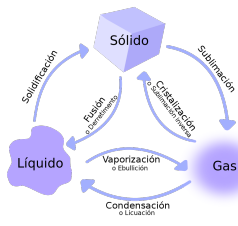
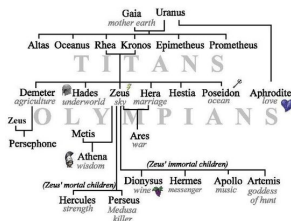
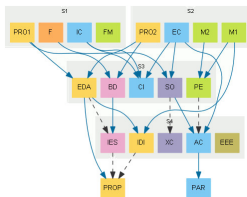
$q_1$	$q_0$	$w_1$	$w_0$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x

- Veuem com representar l'especificació de forma més *human friendly*
  - Graf d'estats



- Introducció
- Estructura general d'un CLS
- **Graf d'estats**
  - Crear graf d'estats a partir de taules de veritat
  - Seguiment de graf d'estats
  - Crear graf d'estats a partir descripció funcional
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

Graf: Parell ordenat  $G := (V, E)$  que comprèn un conjunt  $V$  de vèrtexs o nodes juntament amb un conjunt  $E$  d'arestes o línies, les quals són conjunts de dos elements de  $V$ . [4]



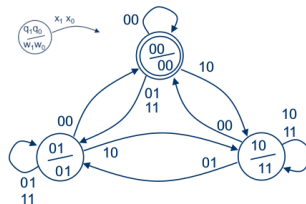
- Representació gràfica del comportament d'un CLS
  - Els nodes del graf representen els estats
    - Cal indicar quin és l'estat inicial
  - Els arcs del graf representen transicions entre estats,
    - Des de cada node sortiran  $2^n$  arcs  
Un per a cada valor possible dels senyals d'entrada
  - El graf ha d'anar acompanyat d'una llegenda
    - Altrament el graf resulta intel·ligible
- **Mateixa informació que les TV's però en un altre format**  
 $Q^+ = \mathcal{G}(X, Q):$

$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
<hr/>	
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
<hr/>	
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
<hr/>	
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x

+

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

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

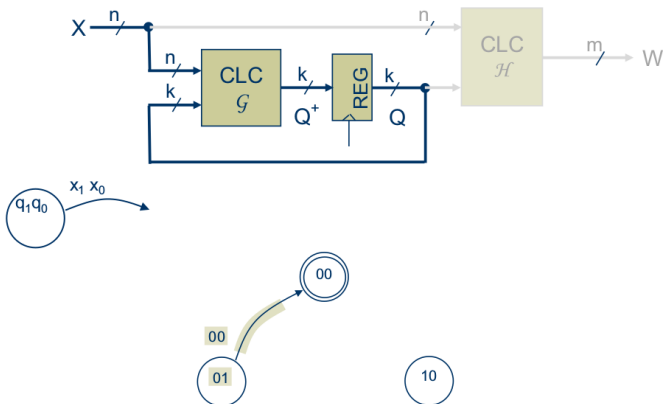


- Introducció
- Estructura general d'un CLS
- **Graf d'estats**
  - Crear graf d'estats a partir de taules de veritat
  - Seguiment de graf d'estats
  - Crear graf d'estats a partir descripció funcional
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Veurem com incorporar la informació de la TV del CLC  $\mathcal{G}$  al graf
  - Els nodes del graf seran els estats (i el "00" l'inicial)
  - Els arcs del graf, transicions entre estats,
    - Des de cada node sortiran  $2^n$  arcs
  - Cal indicar la llegenda del graf ( $n = k = 2$ , falta la sortida)

$Q^+ = \mathcal{G}(X, Q)$  :

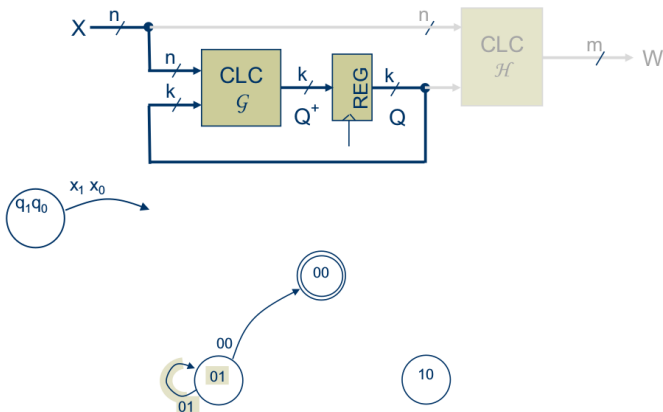
$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x



- Veurem com incorporar la informació de la TV del CLC  $\mathcal{G}$  al graf
  - Els nodes del graf seran els estats (i el "00" l'inicial)
  - Els arcs del graf, transicions entre estats,
    - Des de cada node sortiran  $2^n$  arcs
  - Cal indicar la llegenda del graf ( $n = k = 2$ , falta la sortida)

$$Q^+ = \mathcal{G}(X, Q):$$

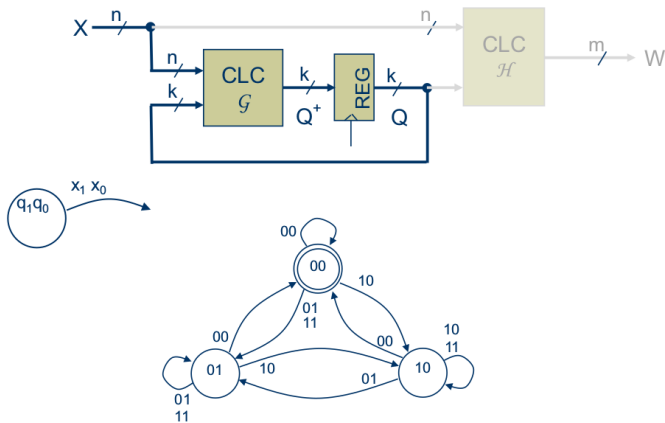
$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x



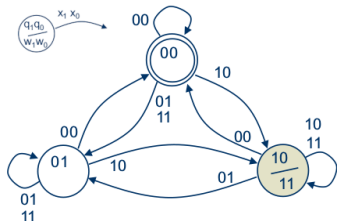
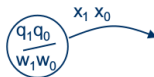
- Veurem com incorporar la informació de la TV del CLC  $\mathcal{G}$  al graf
  - Els nodes del graf seran els estats (i el "00" l'inicial)
  - Els arcs del graf, transicions entre estats,
    - Des de cada node sortiran  $2^n$  arcs
  - Cal indicar la llegenda del graf ( $n = k = 2$ , falta la sortida)

$$Q^+ = \mathcal{G}(X, Q):$$

$q_1 q_0 x_1 x_0$	$q_1^+ q_0^+$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	0 1
<hr/>	
0 1 0 0	0 0
0 1 0 1	0 1
0 1 1 0	1 0
0 1 1 1	0 1
<hr/>	
1 0 0 0	0 0
1 0 0 1	0 1
1 0 1 0	1 0
1 0 1 1	1 0
<hr/>	
1 1 0 0	x x
1 1 0 1	x x
1 1 1 0	x x
1 1 1 1	x x



- Veurem com incorporar la informació de la TV del CLC  $\mathcal{H}$  al graf
  - Com les sortides només depenen de l'estat actual, incorporarem el valor de les sortides als nodes
- Incorporarem les sortides a la llegenda del graf ( $m=2$ )



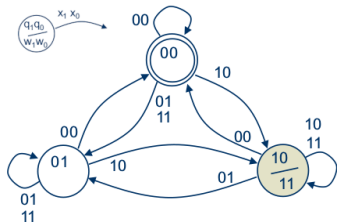
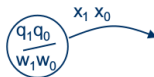
$W = \mathcal{H}(Q)$ :

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

$\Rightarrow$



- Veurem com incorporar la informació de la TV del CLC  $\mathcal{H}$  al graf
  - Com les sortides només depenen de l'estat actual, incorporarem el valor de les sortides als nodes
- Incorporarem les sortides a la llegenda del graf ( $m=2$ )

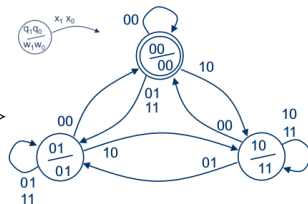


+

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

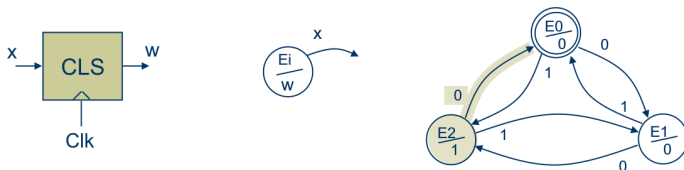
$q_1$	$q_0$	$w_1$	$w_0$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	x	x

$\Rightarrow$



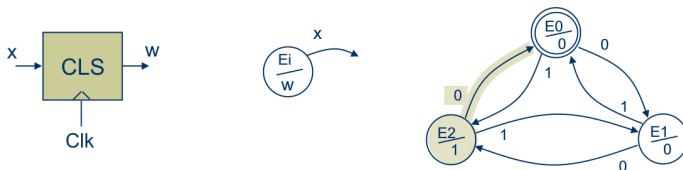
- Introducció
- Estructura general d'un CLS
- **Graf d'estats**
  - Crear graf d'estats a partir de taules de veritat
  - **Seguiment de graf d'estats**
  - Crear graf d'estats a partir descripció funcional
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- A partir del graf d'estats, podem completar el cronograma simplificat que mostra l'evolució de l'estat i de la sortida a mesura que l'entrada canvia de valor.
  - Exemple on  $n = 1, m = 1, k = 2$



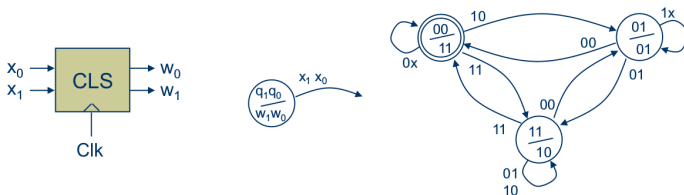
Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Estado	E0	E2	E0	E1	E2	E0	E1	E0	E2	E1					
Entrada x	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
Salida w	0	1	0	0	1	0	0	0	1	0					

- A partir del graf d'estats, podem completar el cronograma simplificat que mostra l'evolució de l'estat i de la sortida a mesura que l'entrada canvia de valor.
  - Exemple on  $n = 1, m = 1, k = 2$



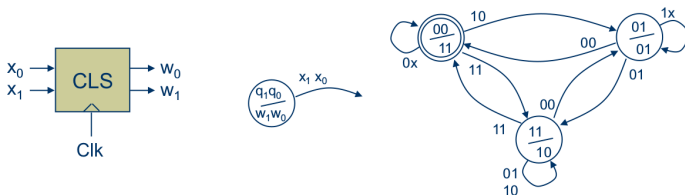
Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Estado	E0	E2	E0	E1	E2	E0	E1	E0	E2	E1	E2	E0	E2	E0	E2
Entrada $x$	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
Salida $w$	0	1	0	0	1	0	0	0	1	0	1	0	1	0	1

- Completeu el cronograma simplificat ( $n = m = k = 2$ )



Núm. Cicle	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Estado $q_1q_0$	00	00	00	11	11										
Entrada $x_1x_0$	00	01	11	10	01	00	00	11	10	01	11	01	00	10	10
Salida $w_1w_0$	11	11	11	10	10										

- Completeu el cronograma simplificat ( $n = m = k = 2$ )



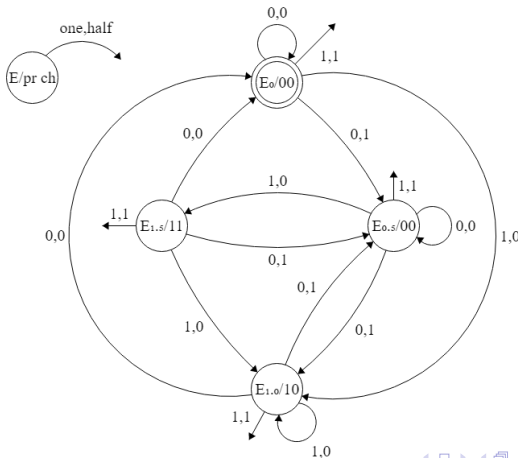
Núm. Cicle	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Estado $q_1 q_0$	00	00	00	11	11	11	01	00	11	11	11	00	00	00	01
Entrada $x_1 x_0$	00	01	11	10	01	00	00	11	10	01	11	01	00	10	10
Salida $w_1 w_0$	11	11	11	10	10	10	01	11	10	10	10	11	11	11	01

- Introducció
- Estructura general d'un CLS
- **Graf d'estats**
  - Crear graf d'estats a partir de taules de veritat
  - Seguiment de graf d'estats
  - Crear graf d'estats a partir descripció funcional
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Dibuixeu el graf d'estats d'un CLS corresponent al mecanisme de control d'una màquina de venda de productes de 1,0€ que admeti monedes de 1€ i de 0,5€ tal que:
  - Entrades:
    - *half*: val 1 si en aquest cicle s'ha introduït una moneda de 0,50€
    - *one*: val 1 si en aquest cicle s'ha introduït una moneda de 1€
    - Podeu assumir que *one* i *half* no poden valer 1 simultàniament
  - Sortides:
    - *product* (pr): valdrà 1 durant un cicle si s'ha rebut un mínim de 1,50€
    - *change* (ch): valdrà 1 durant un cicle si cal retornar canvi (0,50€)
  - Un cop que el CLS notifica que cal servir un producte (i, si cal, retornar canvi), el CLS es disposarà a rebre monedes per a un nou producte.
- Orientacions:
  - Determineu quins estats necessitaríeu i les transicions entre ells en funció de les entrades rebudes
    - Amb  $n$  senyals d'entrada, de cada node han de sortir  $2^n$  arestes
  - No oblideu dibuixar la llegenda i determinar l'estat inicial



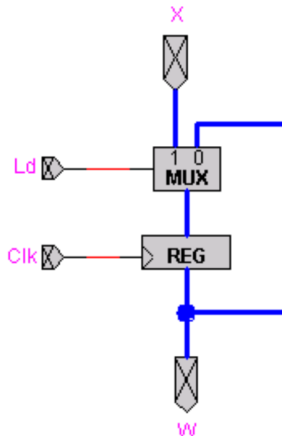
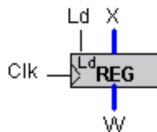
- Nombre d'estats: 4
  - Un per cada quantitat acumulada possible abans de servir el producte
  - 0€, 0.5€, 1.0€ i 1.5€
- Com *one* = 1 i *half* = 1 és impossible, les arestes no porten enlloc



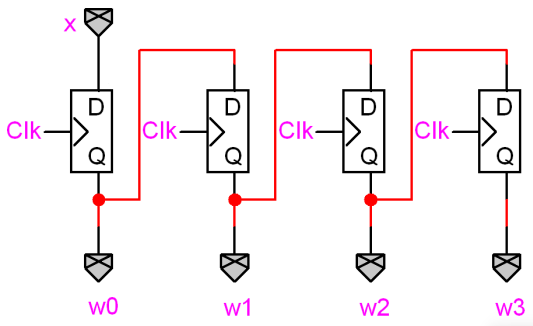
- Introducció
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

# Registre amb senyal de càrrega (*Load*)

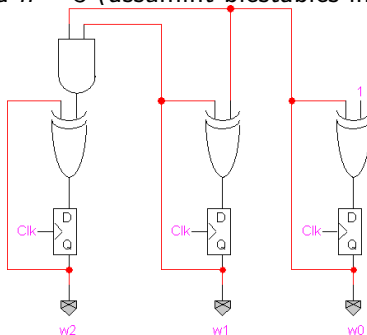
- El REG-*n* actualitza cada cicle el valor que s'hi emmagatzema
- El registre amb senyal de càrrega tindrà un senyal de control (*Ld*)
  - Al produir-se un flanc ascendent de rellotge...
  - Si  $Ld = 1$ , es carregarà al registre l'entrada *X*
  - Si  $Ld = 0$ , es recarregarà el contingut actual del registre



- Emmagatzema els  $n$  darrers bits rebuts per l'entrada
  - Exemple per a  $n = 4$ 
    - Al cicle  $t$ ,  $w_0 = x(t-1)$ ,  $w_1 = x(t-2)$ ,  $w_2 = x(t-3)$ ,  $w_3 = x(t-4)$

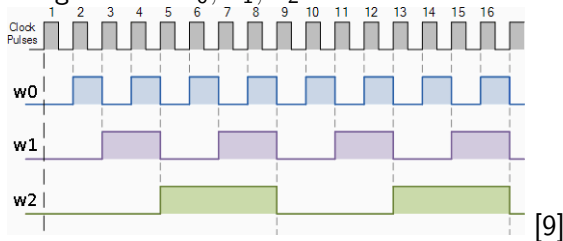


- Genera la seqüència de tots els vectors de  $n$  bits, ordenada seguint la interpretació  $X_u$ , al ritme d'un vector per cicle.
  - En arribar al darrer vector, torna a començar des del primer vector
  - Per a  $n = 2$ , retornarà la seqüència 00, 01, 10, 11, 00, ...
- Implementació per a  $n = 3$  (assumint biestables inicialitzats a 0)



- Com seria la implementació per a  $n = 4$ ?

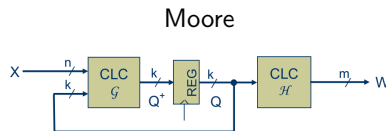
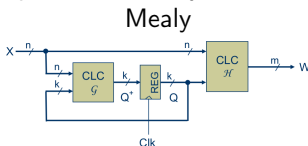
- Com és el cronograma de  $w_0$ ,  $w_1$ ,  $w_2$  ?



- Aquest circuit també rep el nom de divisor de freqüència
- Utilitat pràctica: rellotges de quars
  - Un cristall de quars genera un senyal periòdic a 32.768 ( $2^{15}$ ) Hz
  - Aquest senyal alimenta un comptador de 15 bits
  - El bit de més pes del comptador genera un senyal a 1 Hz
    - Aquest senyal alimenta el mecanisme del rellotge

- Introducció
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- **Conclusions**
- Autoaprenentatge

- Hem introduït els 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
  - CLS amb  $n$  senyals d'entrada,  $m$  de sortida i  $k$  bits d'estat
    - $k$  bits d'estat  $\implies k$  biestables  $\implies$  fins a  $2^k$  estats diferents
  - Un CLC actualitzarà l'estat i l'altre calcularà les sortides
  - Dos possibles dissenys:



- Representarem el comportament dels CLS amb 2 TV's o amb un graf
  - Cal saber fer les conversions entre ambdues representacions
- No oblideu fer l'ET6a a Atenea i l'autoaprenentatge



- Introducció
- Estructura general d'un CLS
- Graf d'estats
- Exemples de CLS's
- Conclusions
- Autoaprenentatge

- Haureu d'estudiar pel vostre compte la següent secció de la documentació de l'assignatura:
  - 6.4 Anàlisi lògic
    - A partir de l'esquema lògic d'un CLS, cal identificar el nombre de bits d'entrada, de sortida i d'estat. A continuació, cal analitzar el CLC que calcula l'estat següent i el que calcula la sortida.

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: <https://commons.wikimedia.org/w/index.php?curid=1058005>.
- [2] [Online]. Available: <http://www.tecno-electronica.com/Su-turno/705-stn-d72sr-su-turno-electronico-de-dos-cifras-via-cable.html>.
- [3] [Online]. Available: <https://alchetron.com/cdn/edward-f-moore-8f7225a0-d71b-4141-ad8b-aea56d5569d-resize-750.jpeg>.
- [4] [Online]. Available: [https://ca.wikipedia.org/wiki/Graf\\_\(matem%C3%A0tiques\)](https://ca.wikipedia.org/wiki/Graf_(matem%C3%A0tiques)).
- [5] [Online]. Available: <https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/pla-destudis>.
- [6] [Online]. Available: [https://www.pinterest.es/pin/208643395223569184/?nic\\_v2=1aprU1lYp](https://www.pinterest.es/pin/208643395223569184/?nic_v2=1aprU1lYp).
- [7] [Online]. Available: <https://www.plano-metro.com/wp-content/uploads/plano-metro-barcelona-1.jpg>.
- [8] [Online]. Available: [https://es.wikipedia.org/wiki/Cambio\\_de\\_estado#/media/Archivo:Estados.svg](https://es.wikipedia.org/wiki/Cambio_de_estado#/media/Archivo:Estados.svg).
- [9] [Online]. Available: [https://www.electronics-tutorials.ws/counter/count\\_1.html](https://www.electronics-tutorials.ws/counter/count_1.html).

# Introducció als Computadors

## Tema 6: Circuits Lògics Seqüencials (CLS's)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC6a.pdf>

Enric Morancho  
([enricm@ac.upc.edu](mailto: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<sup>er</sup> quad.

Presentació publicada sota llicència Creative Commons 4.0