

Introducció als Computadors

Tema 3: Circuits Lògics Combinacionals (CLC)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC3a.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

Periodic table of the elements

period																							
group	1*																18						
1	H																He						
2	Li	Be																B	C	N	O	F	Ne
3	Na	Mg														Al	Si	P	S	Cl	Ar		
4	K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr					
5	Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe					
6	Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn					
7	Fr	Ra	Ac	Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Nh	Fl	Mc	Lv	Ts	Og					
lanthanoid series 6																							
	58	59	60	61	62	63	64	65	66	67	68	69	70	71									
	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu									
actinoid series 7																							
	90	91	92	93	94	95	96	97	98	99	100	101	102	103									
	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr									

*Numbering system adopted by the International Union of Pure and Applied Chemistry (IUPAC).

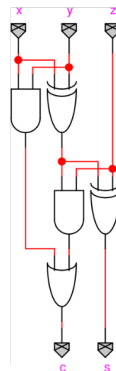
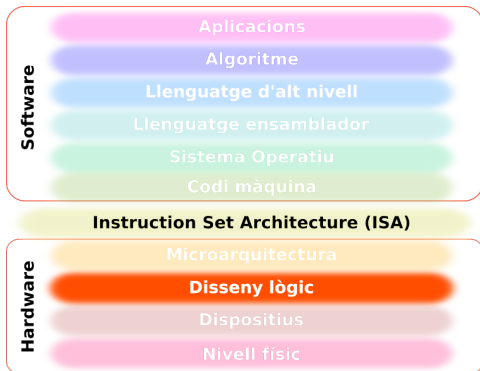
© Encyclopædia Britannica, Inc.

[1]



[2]

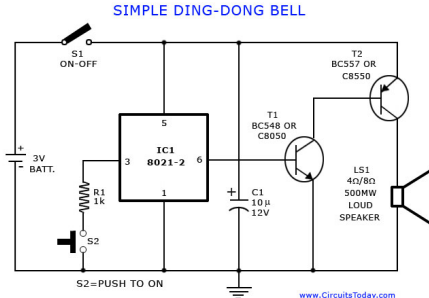
- Introducció
 - Què és un CLC?
 - De la descripció del CLC a la taula de veritat
 - Portes lògiques
 - Connectant CLC's
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània



Full-Adder implementat amb 5 portes lògiques

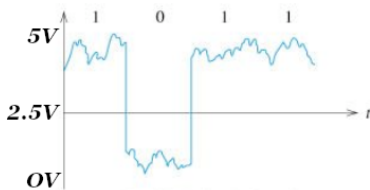
- Introducció
 - Què és un CLC?
 - De la descripció del CLC a la taula de veritat
 - Portes lògiques
 - Connectant CLC's
 - Anàlisi de CLC's
 - Exercicis
 - Conclusions
 - Miscel·lània

- Conjunt de components electrònics connectats
 - Ex: transistors, condensadors, resistències, sensors, altaveu, led, ...
- Té senyals d'entrada
 - Senyal: magnitud física que canvia amb el temps
 - Ex: diferència de potencial elèctric, temperatura, lluminositat, so, ...
- Processa el(s) senyal(s) d'entrada
- Genera senyal(s) de sortida

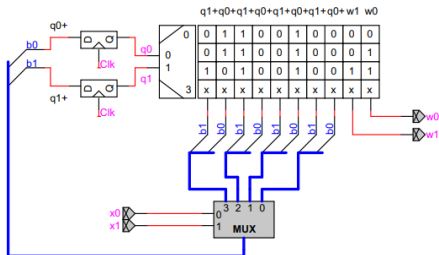


[3]

- Circuit on els senyals s'interpreten digitalment
 - Els senyals es discretitzen (típicament a dos valors lògics)
 - Exemple: si el senyal és una diferència de potencial entre 0V i 5V
 - De 0V a 2,5V: interpretarem que té el valor lògic "0"
 - De 2,5V a 5V: interpretarem que té el valor lògic "1"
 - D'aquesta forma el circuit és immune a petites pertorbacions del senyal causades pel soroll electromagnètic, atenuacions, o d'altres motius
 - Els circuits que no discretitzen senyals es coneixen com a analògics
 - Els components bàsics d'aquests circuits són les portes lògiques



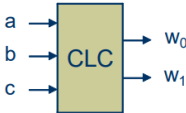
[4]



- Circuit lògic on el valor dels senyals de sortida depèn única i exclusivament del valor actual dels senyals d'entrada
 - És independent del valor previ dels senyals d'entrada
 - "no tenen memòria"
 - Els circuits lògics "amb memòria" (CLS, "Circuits Lògics Seqüencials", tema 6) utilitzen uns components anomenats biestables
- Exemples
 - Són CLC's:
 - sumador de dos nombres de 16 bits, alarma acústica quan $T > 30^{\circ}\text{C}$,...
 - No són CLC's:
 - control monedes a màquina de vending, ...
- Els components bàsics d'aquests circuits són les portes lògiques

- La funcionalitat d'un CLC es pot descriure mitjançant una matriu de bits tal que per a cada possible valor dels senyals d'entrada indiqui el valor dels senyals de sortida
 - Taula de veritat (TV), *Truth table* (TT)
- Mida de la TV d'un CLC amb n senyals d'entrada i m de sortida
 - 2^n **files** $\times m$ **columnes**
 - Una fila per a cada valor possible dels senyals d'entrada
 - Una columna per a cada senyal de sortida
 - Podrem escriure la taula de veritat d'un CLC únicament quan el nombre de variables d'entrada sigui "petit"
- La funcionalitat d'un CLC es pot expressar en llenguatge matemàtic
$$f : \begin{array}{c} \{0, 1\}^n \\ (x_{n-1}, \dots, x_1, x_0) \end{array} \rightarrow \begin{array}{c} \{0, 1\}^m \\ (w_{m-1}, \dots, w_1, w_0) \end{array}$$
 - Direm que els CLC's implementen funcions lògiques

- ❶ Escriurem la capçalera de la taula,
 - Primer el nom de les n variables d'entrada i després de les m de sortida
 - Si no ens ve donat, podem triar qualsevol ordre entre elles
 - Algunes ordenacions poden fer més senzill omplir la TV
- ❷ Amb n variables d'entrada, $x_{n-1}x_{n-2} \dots x_1x_0$, la taula tindrà 2^n files
 - La primera fila serà la fila 0 i la darrera la $2^n - 1$
 - La fila i representa la combinació de valors X tal que $X_u = i$
 - Tot i que $x_{n-1}x_{n-2} \dots x_1x_0$ no tingui res a veure amb un nombre natural representat en binari
 - Exemple: TV d'un CLC amb 3 senyals d'entrada i 2 de sortida
 - El contingut de la TV és la part ombrejada, és a dir, $2^n \times m$ bits



	a	b	c	w_1	w_0
Fila 0 →	0	0	0	0	0
	0	0	1	0	1
	0	1	0	0	1
	0	1	1	1	0
	1	0	0	0	1
	1	0	1	1	0
	1	1	0	1	0
Fila 7 →	1	1	1	1	1

- Introducció
 - Què és un CLC?
 - De la descripció del CLC a la taula de veritat
 - Portes lògiques
 - Connectant CLC's
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània

- Veurem tres exemples de com determinar el contingut de la taula de veritat d'un CLC a partir de la descripció de la seva funcionalitat
 - 1 Majority report
 - 2 Multiplexor
 - 3 Control d'un dipòsit d'aigua

Escriure la TV d'un CLC tal que donats tres vots (sí/no) determini quin és el vot majoritari i si hi ha hagut unanimitat.

- Senyals d'entrada i de sortida?

Escriure la TV d'un CLC tal que donats tres vots (sí/no) determini quin és el vot majoritari i si hi ha hagut unanimitat.

- Senyals d'entrada i de sortida?
 - 3 senyals d'entrada d'un bit (els 3 vots)
 - 2 senyals de sortida d'un bit (el vot majoritari i si és unànime)

Escriure la TV d'un CLC tal que donats tres vots (sí/no) determini quin és el vot majoritari i si hi ha hagut unanimitat.

- Senyals d'entrada i de sortida?
 - 3 senyals d'entrada d'un bit (els 3 vots)
 - 2 senyals de sortida d'un bit (el vot majoritari i si és unànime)
- Mida de la taula de veritat?

Escriure la TV d'un CLC tal que donats tres vots (sí/no) determini quin és el vot majoritari i si hi ha hagut unanimitat.

- Senyals d'entrada i de sortida?
 - 3 senyals d'entrada d'un bit (els 3 vots)
 - 2 senyals de sortida d'un bit (el vot majoritari i si és unànime)
- Mida de la taula de veritat?
 - $2^3 = 8$ files (una per a cada combinació possible dels senyals d'entrada)
 - 2 columnes (una per a cada senyal de sortida)

Escriure la TV d'un CLC tal que donats tres vots (sí/no) determini quin és el vot majoritari i si hi ha hagut unanimitat.

- Senyals d'entrada i de sortida?
 - 3 senyals d'entrada d'un bit (els 3 vots)
 - 2 senyals de sortida d'un bit (el vot majoritari i si és unànime)
- Mida de la taula de veritat?
 - $2^3 = 8$ files (una per a cada combinació possible dels senyals d'entrada)
 - 2 columnes (una per a cada senyal de sortida)
- Encapsulat del CLC
 - En aquest cas, totes les entrades són equivalents
 - Ara bé, cal especificar quina és cada sortida perquè són diferents



- Taula de veritat
 - Codificació dels senyals: convencional (0=no, 1=sí)

- Taula de veritat
 - Codificació dels senyals: convencional (0=no, 1=sí)

v_0	v_1	v_2	resultat	unànim
-------	-------	-------	----------	--------

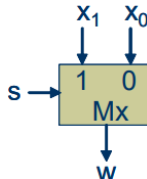
- Taula de veritat
 - Codificació dels senyals: convencional (0=no, 1=sí)

v_0	v_1	v_2	resultat	unànime
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- En aquest cas ens especifiquen la funcionalitat com a pseudocodi:

```
if (s==0) w=x0; else w=x1;
```

- Hauria de ser trivial deduir quins són els senyals d'entrada i de sortida
 - 3 senyals d'entrada: s , x_0 i x_1
 - Cal identificar-les perquè no són intercanviables
 - Marquem on han d'anar a parar x_1 (1) i x_0 (0)
 - 1 senyal de sortida: w



- Mida de la taula de veritat:
 - 2^3 files \times 1 columna
- Aquest CLC serà molt utilitzat als nostres circuits
 - Selecciona una entrada entre dues

- Taula de veritat
 - En aquest cas, dependent de com ordenem les variables d'entrada, omplir la taula de veritat pot resultar més o menys senzill

- Taula de veritat
 - En aquest cas, dependent de com ordenem les variables d'entrada, omplir la taula de veritat pot resultar més o menys senzill

S	x_0	x_1	W
-----	-------	-------	-----

x_0	x_1	S	W
-------	-------	-----	-----

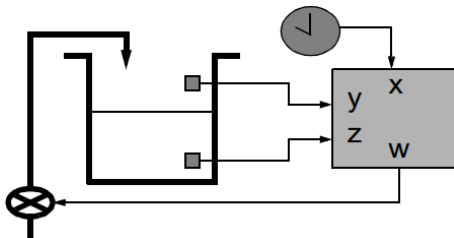
- Taula de veritat
 - En aquest cas, dependent de com ordenem les variables d'entrada, omplir la taula de veritat pot resultar més o menys senzill

s	x ₀	x ₁	w
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

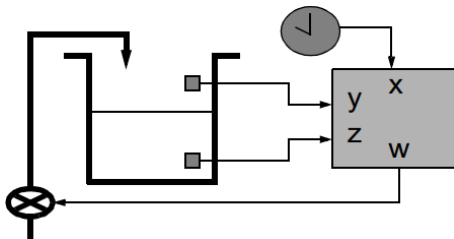
x ₀	x ₁	s	w
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Exemple 3: Control d'un dipòsit

Sea un CLC con 3 entradas y 1 salida para controlar el llenado automático de un depósito de agua. La entrada x viene de un reloj y vale 1 durante el día y 0 durante la noche. Las entradas y y z vienen de dos sensores colocados dentro del depósito, en la parte superior e inferior respectivamente. Cada señal indica con el valor 1 qué el sensor está cubierto de agua y con 0 que no lo está. La salida del circuito controla una bomba de agua, cuando w vale 1 se bombea agua de un pozo para llenar el depósito y cuando vale 0 no. El agua del depósito se va consumiendo de forma irregular durante las 24 horas del día. Se desea que siempre haya agua en el depósito, pero a poder ser, accionando la bomba por la noche.



- L'esquema de l'enunciat ens indica nombre entrades i sortides



- 3 senyals d'entrada: x , y i z
- 1 senyal de sortida: w
- Mida de la taula de veritat:
 - 2^3 files \times 1 columna

- Podem començar pels casos extrems
 - Quan el dipòsit és plé ($y = 1, z = 1$, sigui de nit o de dia)
 - No bombejarem aigua
 - Quan el dipòsit és buit ($y = 0, z = 0$, sigui de nit o de dia)
 - Bombejarem aigua

x	y	z	w
0	0	0	1
0	0	1	
0	1	0	
0	1	1	0
1	0	0	1
1	0	1	
1	1	0	
1	1	1	0

Exemple 3: Control d'un dipòsit



- Quan el dipòsit és a mitges ($y = 0, z = 1$)
 - Quan sigui de dia ($x = 1$)
 - No bombejarem aigua
 - Quan sigui de nit ($x = 0$)
 - Bombejarem aigua

x	y	z	w
0	0	0	1
0	0	1	1
0	1	0	
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	
1	1	1	0

Exemple 3: Control d'un dipòsit



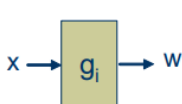
- Les combinacions on $y = 1, z = 0$ són impossibles
 - Si el sensor y indica aigua, z també n'ha d'indicar
 - Llevat malfuncionament dels sensors... assumirem que no és el cas.
 - Però han de ser presents a la TV
- Omplirem aquestes posicions de la TV amb el símbol x
 - x indica *Don't care*
 - Això ens donarà més llibertat a l'hora de sintetitzar el CLC
 - Acabarem posant-hi el valor que més convingui per simplificar el CLC

x	y	z	w
0	0	0	1
0	0	1	1
0	1	0	x
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	x
1	1	1	0

- Introducció
 - Què és un CLC?
 - De la descripció del CLC a la taula de veritat
 - Portes lògiques
 - Connectant CLC's
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània

- Són els CLC's més senzills
 - Amb una entrada d'un bit i una sortida d'un bit
 - Amb dues entrades d'un bit i una sortida d'un bit
- Seran la base per a construir CLC's més complexes
 - **Donada qualsevol taula de veritat, sempre podrem construir un circuit format únicament amb portes lògiques que la satisfaci.**
- Com les entrades i sortides són binàries, podem analitzar **tots** els CLC's possibles amb 1 i amb 2 entrades
 - Això no es pot fer amb les funcions matemàtiques de variable real perquè el domini d'entrada és infinit
 - Amb les funcions lògiques ho podrem fer sempre que el nombre de variables sigui petit
 - n variables lògiques d'entrada $\implies 2^n$ combinacions possibles

- Només hi ha 4 CLC's possibles



$$w = g_i(x)$$

x	g_3	g_2	g_1	g_0
0	1	1	0	0
1	1	0	1	0

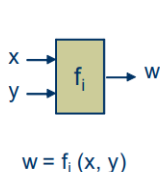


Not

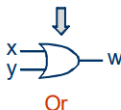
$$w = !x$$

- g_0, g_1, g_3 no tenen interès
 - g_0, g_3 retornen un valor constant
 - g_1 retorna el mateix valor que tenim a l'entrada
- g_2 sí, inverteix el valor que tenim a l'entrada
 - En direm porta NOT
 - Algebraicament la representarem amb el símbol !
 - Als circuits la representarem amb un triangle i un cercle

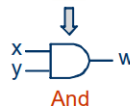
- Dels 16 CLC's possibles (2^{2^n} , on $n = 2$), considerarem:



x	y	f_{15}	f_{14}	f_{13}	f_{12}	f_{11}	f_{10}	f_9	f_8	f_7	f_6	f_5	f_4	f_3	f_2	f_1	f_0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0



$$w = x + y$$



$$w = x \cdot y$$

- f_{14} : retorna 1 \iff alguna entrada val 1, en direm OR (OR-2)
 - Algebràicament tindrà el símbol $+$ i als circuits el símbol indicat
- f_8 : retorna 1 \iff les dues entrades valen 1, en direm AND (AND-2)
 - Algebràicament tindrà el símbol \cdot i als circuits el símbol indicat

- Veurem que qualsevol CLC es pot implementar utilitzant únicament portes NOT, portes AND-2 i portes OR-2.
 - Independentment del nombre de senyals d'entrada i de sortida del CLC
- De moment assumirem que processen les entrades en temps 0
 - Properament considerarem el retard que introdueixen a la sortida

Not



$$w = !x$$
$$(w = \overline{x})$$
$$(w = x')$$

x	w
0	1
1	0

And-2



$$w = x y$$
$$(w = x \cdot y)$$
$$(w = x \&\& y)$$

x	y	w
0	0	0
0	1	0
1	0	0
1	1	1

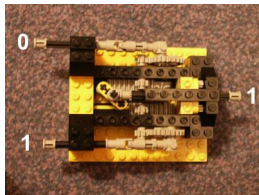
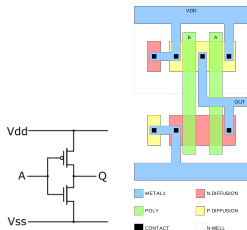
Or-2



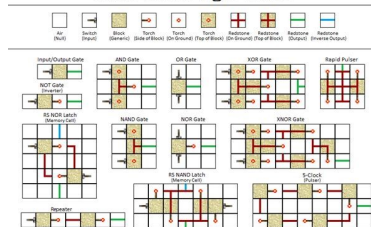
$$w = x + y$$
$$(w = x + y)$$
$$(w = x \parallel y)$$

x	y	w
0	0	0
0	1	1
1	0	1
1	1	1

- No ens toca a IC :-)
- A IC construïrem circuits assumint que les portes lògiques es comporten com indiquen les seves taules de veritat
 - IC és independent de la implementació física de les portes
- Però, com s'implementen?
 - Electrònicament, per exemple, amb tecnologia CMOS [5]
 - També es poden implementar amb LEGO [6] o Minecraft [7] :-)



MineCraft Logic Gates



Humor amb portes lògiques?



[8]

OR gate



AND gate



COL gate



BILL gates

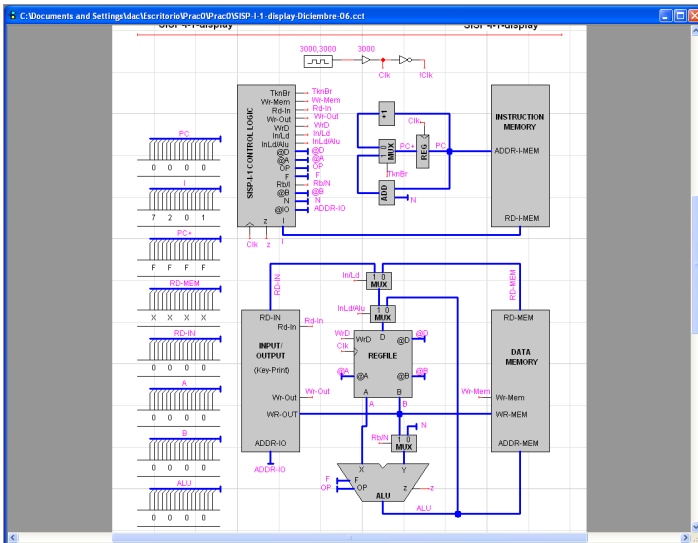


[9]

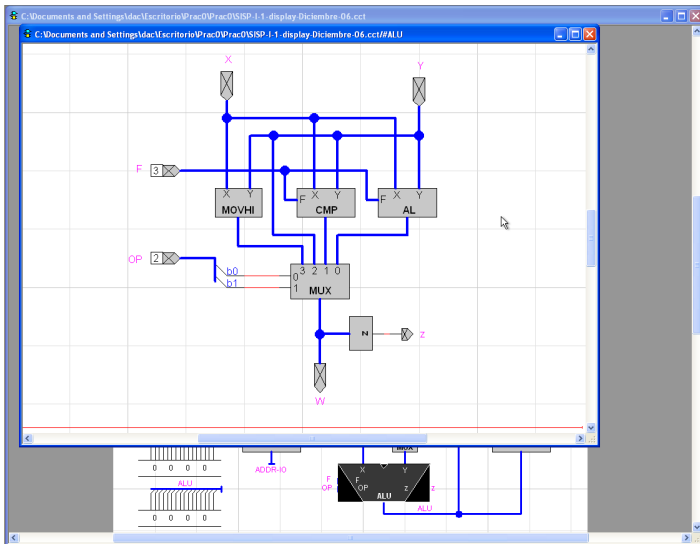
- Introducció
 - Què és un CLC?
 - De la descripció del CLC a la taula de veritat
 - Portes lògiques
 - **Connectant CLC's**
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània

- Al curs dissenyarem un computador senzill on tots els seus CLC's estaran fets amb portes lògiques AND-2, OR-2 i NOT
- Estratègia que seguirem:
 - Primer dissenyarem CLC's senzills directament amb portes lògiques
 - Per exemple, multiplexors, half-adders,...
 - Un cop verificats, els incorporarem a la nostra biblioteca de circuits
 - Després dissenyarem CLC's més complexes a partir dels CLC's acabats de dissenyar i de les portes lògiques bàsiques
 - Per exemple, sumadors, comparadors, ...
 - Un cop verificats, els incorporarem a la nostra biblioteca de circuits
 - Aquest procés el repetirem els nivells que facin falta fins a arribar a la funcionalitat final
- Aquesta estratègia és imprescindible en circuits "grans" i en facilita molt l'anàlisi, la depuració i la divisió del treball

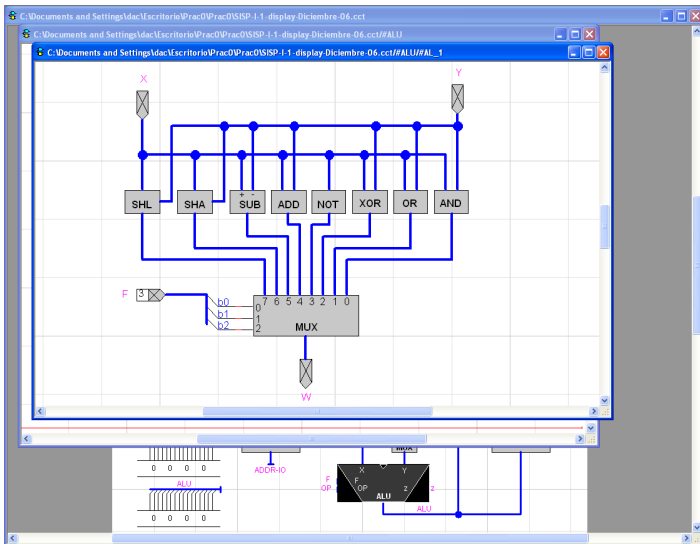
- Esquema final del computador que dissenyarem



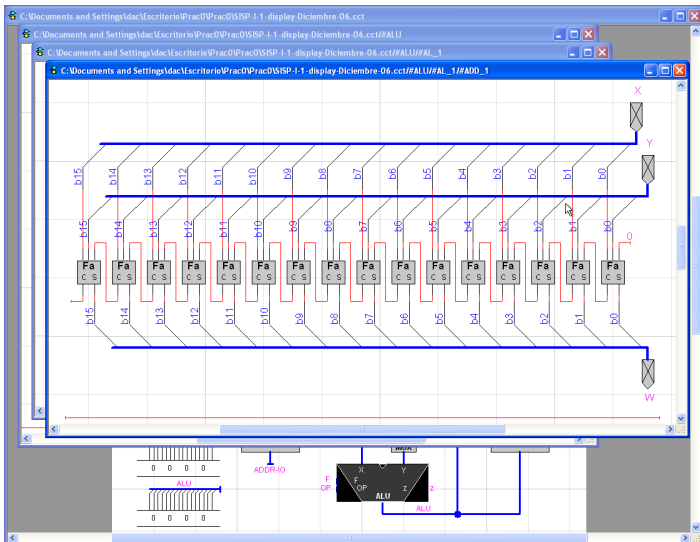
- Entrem a bloc ALU



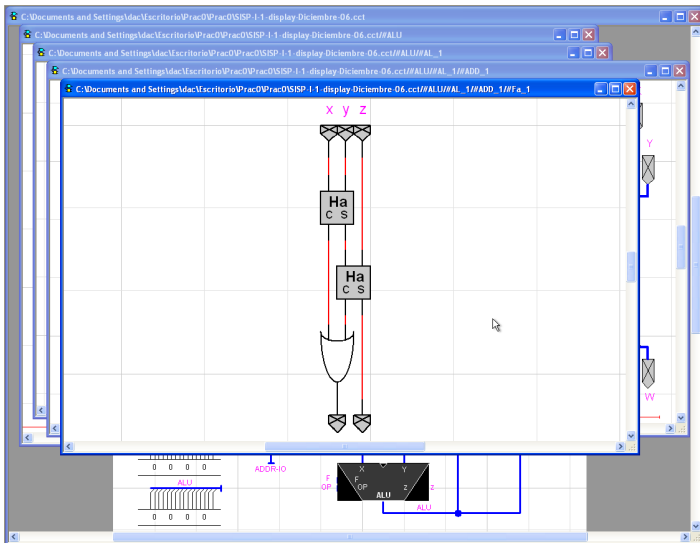
- Entrem a bloc ALU



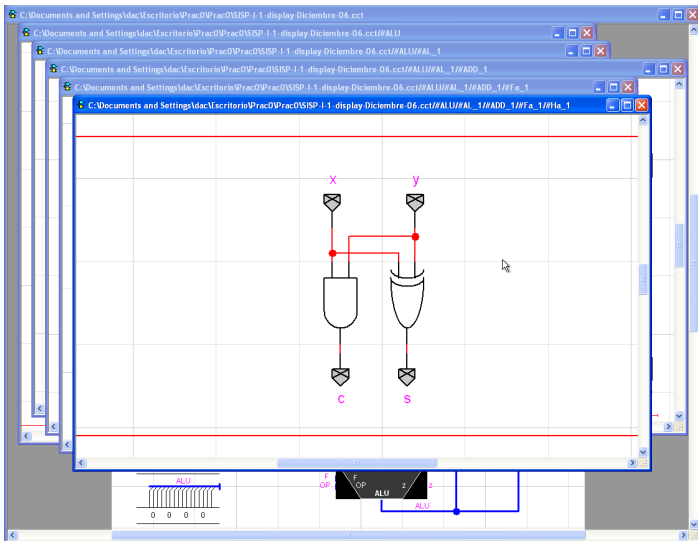
- Entrem al bloc ADD



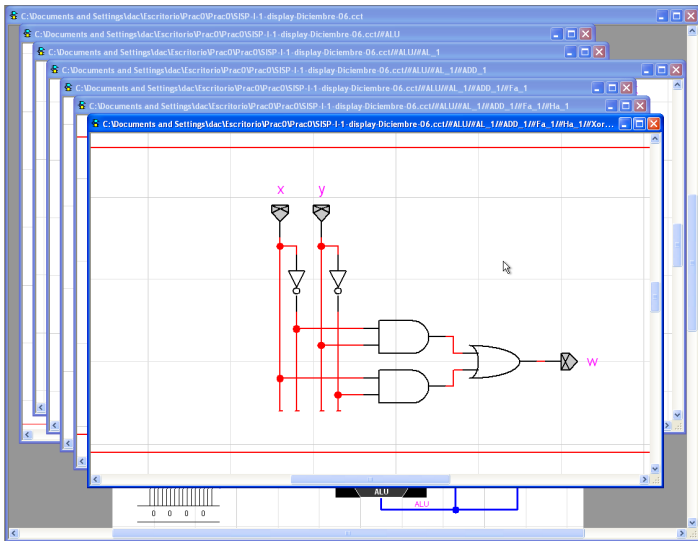
- Entrem al bloc FA (*Full-adder*)



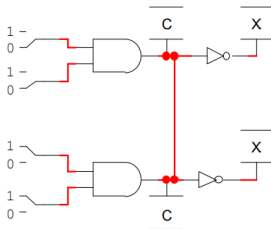
- Entrem al bloc HA (*Half-adder*)



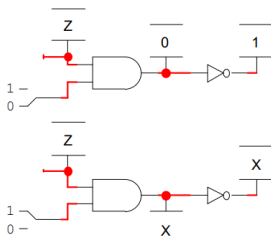
- Entrem al bloc XOR (*Exclusive OR*)



- ❶ No podem connectar directament dos o més sortides d'un CLC
 - Provocarem un curtcircuit
 - Logic Works mostra el símbol C
- ❷ Cal definir el valor de totes les entrades d'un CLC
 - Si no ho fem, tindrem una entrada en alta impedància
 - Logic Works mostra el símbol Z
- ❸ No pot haver cap cicle (camí tancat)
 - Si n'existeix, pot passar que aparegui un valor inestable
 - Si n'existeix, pot passar que aparegui un valor estable
 - Deixaria de ser un CLC perquè el circuit tindria "memòria"



- Al connectar les sortides de les portes AND-2 hem creat un curtcircuit
 - Logic Works mostra una C al *probe* sobre el curtcircuit
- A la sortida de les NOT, Logic Works mostra valor desconegut (X)



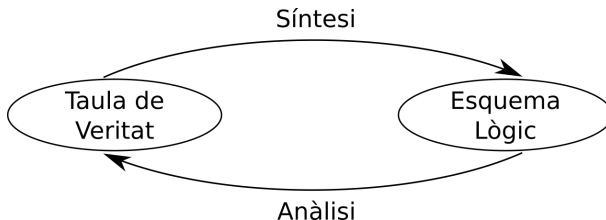
- Hem deixat una entrada sense definir
 - Logic Works mostra una Z al *probe* sobre el cable
- La Z es propagarà a la sortida de la AND-2 en funció del valor de l'altra entrada
 - A la porta AND-2, les entrades amb valor 0 són dominants
 - Però als dos casos el circuit és incorrecte



- Existència d'un cicle que provoca un valor inestable

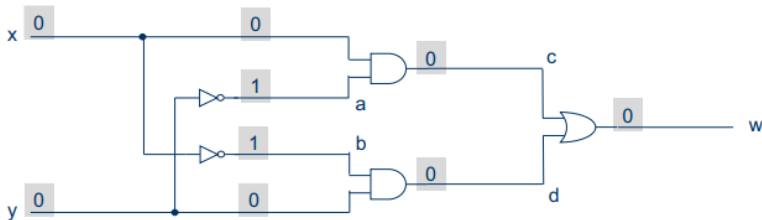
- Introducció
- Anàlisi de CLC's
 - Omplint la TV per files
 - Omplint la TV per columnes
- Exercicis
- Conclusions
- Miscel·lània

- Donat l'esquema lògic d'un CLC, determinar la seva taula de veritat
 - El pas contrari rep el nom de síntesi
 - Donada una TV, elaborar un CLC que es comporti com indica la taula
- Veurem dues formes de fer l'anàlisi d'un CLC
 - Per files
 - Per columnes



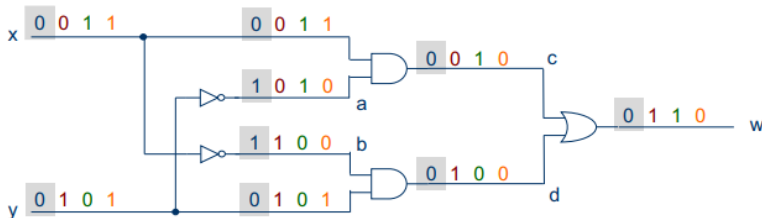
- Introducció
- Anàlisi de CLC's
 - Omplint la TV per files
 - Omplint la TV per columnes
- Exercicis
- Conclusions
- Miscel·lània

- Considerem l'exemple de les figures 3.15 i 3.16 de la documentació
- Aquest CLC té dos senyals d'entrada
 - La seva TV tindrà 2^2 files
- Considerem la combinació de valors 00 als senyals d'entrada
 - Els propaguem fins a la sortida considerant les TV's de les portes
 - Obtenim el valor de la sortida corresponent a la primera fila de la TV



x	y	w
0	0	0
0	1	
1	0	
1	1	

- Repetim el procés amb la resta de files de la TV



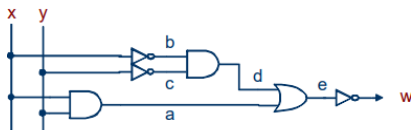
x	y	w
0	0	0
0	1	1
1	0	1
1	1	0



- La TV indica que aquest circuit es comporta com la porta XOR
 - XOR = eXclusive-OR

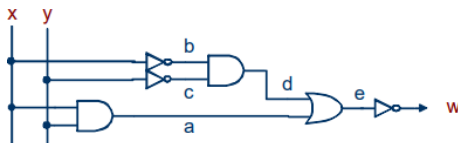
- Introducció
- Anàlisi de CLC's
 - Omplint la TV per files
 - Omplint la TV per columnes
- Exercicis
- Conclusions
- Miscel·lània

- Considerem l'exemple de les figures 3.17 i 3.18 de la documentació
- Aquest CLC té dos senyals d'entrada
 - La seva TV tindrà 2^2 files
- Considerarem totes les sortides internes dels CLC's del circuit
 - A la TV afegirem una columna per a cada sortida interna
 - Podem etiquetar-la amb un nom o l'expressió lògica equivalent
 - Omplirem els valors d'entrada del CLC amb els valors estàndar



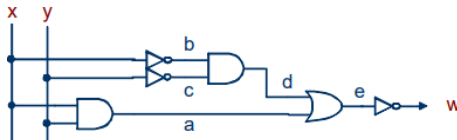
x	y	x·y a	!x b	!y c	b·c d	a+d e	!e w
0	0						
0	1						
1	0						
1	1						

- Ara, omplirem la TV de columna en columna a mesura que totes les entrades dels CLC's interns siguin conegudes
 - Comencem amb la columna *a*, continueu amb la resta de columnes...



x	y	x·y a	!x b	!y c	b·c d	a+d e	!e w
0	0	0					
0	1	0					
1	0	0					
1	1	1					

- En acabar, podrem extreure la TV



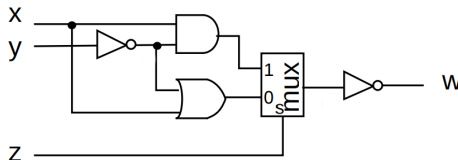
x	y	x·y	!x	!y	b·c	a+d	!e
0	0	0	1	1	1	1	0
0	1	0	1	0	0	0	1
1	0	0	0	1	0	0	1
1	1	1	0	0	0	1	0

x	y	w
0	0	0
0	1	1
1	0	1
1	1	0

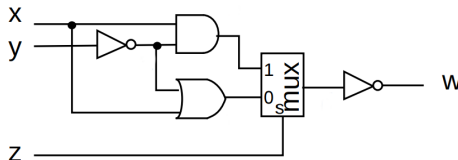
- La TV indica que aquest circuit també es comporta com la porta XOR
 - Hem vist dos CLC's diferents que tenen la mateixa taula de veritat

- Introducció
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània

- Deduïu la taula de veritat del següent circuit (mux és el CLC de l'exemple 2, slide 15)



- Deduïu la taula de veritat del següent circuit (mux és el CLC de l'exemple 2, slide 15)

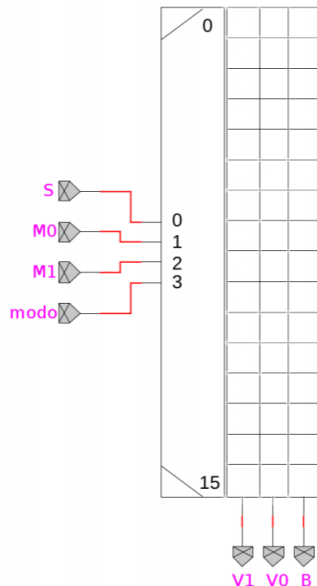


x	y	z	w
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Escribid el contenido de la ROM del CLC que controla el motor del limpiaparabrisas de un coche. El CLC tiene tres señales de entrada (*modo* -1 bit-, *M* -2 bits- y *S* -1 bit-) y dos de salida (*V* -2 bits- y *B* -1 bit-). *V* indica la velocidad de barrido de la escobilla; los valores 00, 01, 10 y 11 codifican, respectivamente, las velocidades *nula*, *lenta*, *rápida* y *muy rápida*. *B* indica si debe activarse la bomba de agua del limpiaparabrisas (el valor 0 codifica que no debe activarse y el valor 1 que debe hacerlo).

En modo automático (*modo*=0), la entrada *S* indica si el parabrisas está limpio (*S*=0) o sucio (*S*=1). Mientras esté sucio, la bomba de agua debe activarse y la escobilla debe barrer a velocidad *lenta*. La entrada *M* codifica la intensidad de lluvia en este momento (los valores 00, 01, 10 y 11 codifican, respectivamente, *sin lluvia*, *moderada*, *intensa* y *muy intensa*). La salida *V* tomará el valor *nula*, *lenta*, *rápida* y *muy rápida* cuando la intensidad de lluvia sea, respectivamente, *sin lluvia*, *moderada*, *intensa* y *muy intensa*. Nos garantizan que las entradas nunca mostrarán simultáneamente presencia de suciedad y de lluvia.

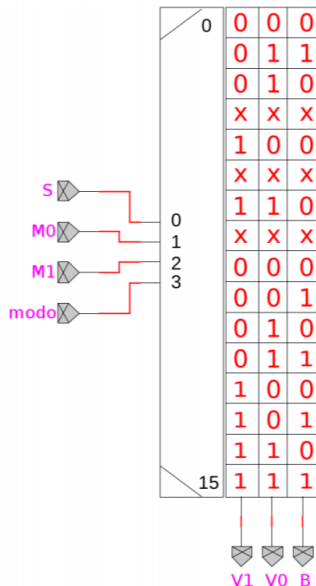
En modo manual (*modo*=1), la entrada *S* indica que el conductor ha accionado el pulsador de la bomba de agua del parabrisas (*S*=1); por tanto, la orden debe transmitirse a la bomba de agua. La entrada *M* indica la velocidad de barrido seleccionada por el conductor (los valores 00, 01, 10 y 11 codifican, respectivamente, las velocidades *nula*, *lenta*, *rápida* y *muy rápida*); esta selección debe transmitirse al motor.



Escribid el contenido de la ROM del CLC que controla el motor del limpiaparabrisas de un coche. El CLC tiene tres señales de entrada (*modo* -1 bit-, *M* -2 bits- y *S* -1 bit-) y dos de salida (*V* -2 bits- y *B* -1 bit-). *V* indica la velocidad de barrido de la escobilla; los valores 00, 01, 10 y 11 codifican, respectivamente, las velocidades *nula*, *lenta*, *rápida* y *muy rápida*. *B* indica si debe activarse la bomba de agua del limpiaparabrisas (el valor 0 codifica que no debe activarse y el valor 1 que debe hacerlo).

En modo automático (*modo*=0), la entrada *S* indica si el parabrisas está limpio (*S*=0) o sucio (*S*=1). Mientras esté sucio, la bomba de agua debe activarse y la escobilla debe barrer a velocidad *lenta*. La entrada *M* codifica la intensidad de lluvia en este momento (los valores 00, 01, 10 y 11 codifican, respectivamente, *sin lluvia*, *moderada*, *intensa* y *muy intensa*). La salida *V* tomará el valor *nula*, *lenta*, *rápida* y *muy rápida* cuando la intensidad de lluvia sea, respectivamente, *sin lluvia*, *moderada*, *intensa* y *muy intensa*. Nos garantizan que las entradas nunca mostrarán simultáneamente presencia de suciedad y de lluvia.

En modo manual (*modo*=1), la entrada *S* indica que el conductor ha accionado el pulsador de la bomba de agua del parabrisas (*S*=1); por tanto, la orden debe transmitirse a la bomba de agua. La entrada *M* indica la velocidad de barrido seleccionada por el conductor (los valores 00, 01, 10 y 11 codifican, respectivamente, las velocidades *nula*, *lenta*, *rápida* y *muy rápida*); esta selección debe transmitirse al motor.



- Introducció
- Anàlisi de CLC's
- Exercicis
- **Conclusions**
- Miscel·lània

- A un CLC, el valor dels senyals de sortida depèn única i exclusivament del valor actual dels senyals d'entrada
- La funcionalitat d'un CLC amb n senyals d'entrada i m de sortida es pot expressar amb una taula de veritat amb 2^n files \times m columnes
- Els CLC's més senzills, les portes lògiques AND-2, OR-2 i NOT, seran la base que utilitzarem per construir CLC's més complexes
 - Veurem que són prou potents per a poder construir qualsevol CLC
 - Dissenyarem els CLC's modularment
 - Al connectar CLC's cal respectar 3 regles
 - No crear curtcircuits, definir totes les entrades, no crear cicles
- L'anàlisi de l'esquema lògic d'un CLC consisteix en determinar la seva taula de veritat
 - Hem vist dues estratègies: per files i per columnes
 - CLC's diferents poden tenir la mateixa taula de veritat
- **No oblideu respondre l'ET3a a Atenea abans de la propera classe**

- Introducció
- Anàlisi de CLC's
- Exercicis
- Conclusions
- Miscel·lània

Símbols alternatius de les portes lògiques



Logic function	American (MIL/ANSI) Symbol	British (BS3939) Symbol	Common German Symbol	International Electrotechnical Commission (IEC) Symbol
	IN OUT	IN OUT	IN OUT	IN OUT
Buffer				
Inverter (NOT gate)				
2-input AND gate				
2-input NAND gate				
2-input OR gate				
2-input NOR gate				
2-input EX-OR gate				
2-input EX-NOR gate				

[10]

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://www.britannica.com/science/periodic-table>.
- [2] [Online]. Available: <https://www.pinterest.es/pin/170010954663237098/>.
- [3] [Online]. Available: <http://www.CircuitsToday.com/>.
- [4] [Online]. Available: https://es.wikipedia.org/wiki/Archivo:Imagen_4.png.
- [5] [Online]. Available: https://www.cs.uaf.edu/2011/fall/cs441/lecture/09_05_semiconductors.html.
- [6] [Online]. Available: <https://www.randomwraith.com/logic.html>.
- [7] [Online]. Available: <https://minecraft.wonderhowto.com/news/redstone-logic-gates-mastering-fundamental-building-blocks-for-creating-game-machines-0135063/>.
- [8] [Online]. Available: www.humorthatworks.com/database/android-logo-logic-gates.
- [9] [Online]. Available: <https://me.me/i/or-gate-out-and-gate-out-col-gate-colgate-avi-2550614>.
- [10] [Online]. Available: https://www.nutsvolts.com/magazine/article/understanding_digital_buffer_gate_and_ic_circuits_part_1.

Introducció als Computadors

Tema 3: Circuits Lògics Combinacionals (CLC)

<http://personals.ac.upc.edu/enricm/Docencia/IC/IC3a.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