UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Generació de funcions lògiques mitjançant descodificadors binaris amb eixides actives a nivell alt

Cognoms, nom	Martí Campoy, Antonio (amarti@disca.upv.es)
Departament	Informàtica de Sistemes i Computadors
Centre	Universitat Politècnica de València

1 Resum de les idees clau

En aquest article es presentarà la utilització de descodificadors binaris amb eixides actives a nivell alt per a la generació de funcions lògiques. Són moltes les formes de dissenyar una funció lògica, i una de les més senzilles és la utilització del bloc combinacional conegut com a descodificador binari. Per a poder adquirir els coneixements i habilitats presentades en aquest article, has de tenir els coneixements previs presentats en la taula 1, encara que durant el text, els recordarem un poc.

Taula 1 Coneixements previs

Coneixements previs
1. Què és una funció lògica i la seua aritat
2. Tipus i taules de veritat de portes lògiques
3. Formes de representar una funció lògica: taula de veritat, formes canòniques i expressions algebraiques
4. Funcionament d'un descodificador binari
5. Circuit intern d'un descodificador binari

2 Objectius

Una vegada acabes de llegir aquest article docent i reproduïts els exemples presentats, hauràs de ser capaç **d'implementar** una funció lògica mitjançant l'ús de descodificadors binaris amb eixides actives a nivell **alt**.

A més a més, la implementació de la funció lògica podrà prendre com a punt de partida diferents representacions de la funció, com la taula de veritat o una forma canònica, per la qual cosa seràs capaç de **traduir** des d'una representació a una altra.

Finalment, i atenent criteris de simplificació de circuits, podràs **triar** el tipus de porta lògica més adequada.

3 Introducció

En primer lloc, una breu descripció dels conceptes previs més importants per a poder assolir els objectius proposats en aquest article. Aquestes descripcions poden ampliar-se consultant la bibliografia proposada al final del document.

- Funció lògica: expressió formal del comportament d'un circuit digital. L'aritat d'una funció lògica és el nombre de variables d'entrada.

- Taula de veritat: representació única en forma de taula d'una funció lògica.
- Formes canòniques: representació única com a suma de productes o com a producte de sumes d'una funció lògica.
- Expressió algebraica: combinació de variables i operadors lògics per a expressar una funció lògica.
- Porta lògica: circuit digital que implementa una funció lògica bàsica.
- Circuit o funció combinacional: circuit en què les eixides en un instant de temps depenen exclusivament de les entrades en el mateix instant de temps.
- Descodificador binari: circuit combinacional, amb m entrades binàries i $n=2^m$ eixides binàries. Les eixides s'activen de forma exclusiva, és a dir, només s'activa una d'elles en un instant concret.
- La funció realitzada per un descodificador binari consisteix a activar l'eixida d'ordre j que correspon amb la codificació binària de les seues entrades. La Figura 1 presenta el símbol lògic d'un descodificador binari d'm a $n=2^m$ amb eixides actives a nivell **alt**.
- Una eixida activa a nivell alt prendrà valor un quan estiga activada, i valor zero quan no estiga activada.

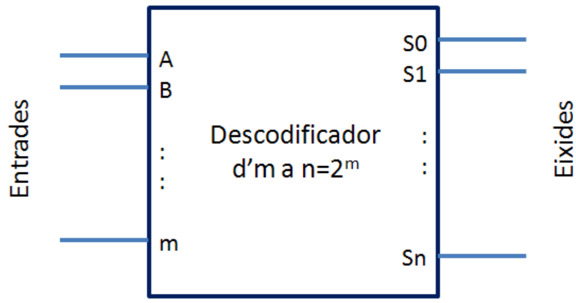

Figura 1 Símbol lògic d'un descodificador d'm a n amb eixides actives a nivell alt

Per comprovar si el funcionament d'aquest descodificador és clar, ens farem un parell de preguntes. Suposem un descodificador binari de 2 a 4 amb eixides actives a nivell alt. Les entrades s'anomenen B i A, sent A la de menor pes, i les eixides reben el nom de S0, S1, S2 i S3. Si els valors de les entrades són B=1 i A=0, el valor de les eixides S0, S1, S2 i S3 és:

Per favor, pensa quina és la resposta abans de mirar la solució[1]

Provem-ho una altra vegada. Si els valors de les entrades són B=0 i A=1, el valor de les eixides S0, S1, S2 i S3 és:

Per favor, pensa quina és la resposta abans de mirar la solució[2]

---

[1] El valor de les eixides és S0=0, S1=0, S2=1, S3=0, perquè BA=10 es correspon amb el valor binari 2, per la qual cossa s'activa l'eixida S2 amb valor 1 (alt).

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

4 Generació de funcions

En aquest apartat veurem, primer, el significat algebraic de les eixides del descodificador binari amb eixides actives a nivell alt. En segon lloc, recordarem breument que una funció lògica pot crear-se a partir de la forma canònica disjuntiva, coneguda com la suma dels seus minitermes. Finalment, unint les dues idees prèvies, usarem portes lògiques per a generar una funció utilitzant descodificadors binaris amb eixides actives a nivell alt.

4.1 Significat de les eixides del descodificador binari

La implementació interna d'un descodificador binari amb eixides actives a nivell alt és molt senzilla. Per a cada una de les eixides es realitza un circuit que l'activarà (posant un 1) si les entrades prenen el valor corresponent. Per exemple, per a un descodificador de 2 a 4, les eixides es corresponen amb les expressions algebraiques mostrades en l'Equació 1, que coincideixen, a més a més, amb les expressions dels minitermes[3].

Ara que coneixem que les eixides d'un descodificador binari corresponen amb la implementació de cadascun dels minitermes, podem incloure aquesta informació en el símbol lògic, que es mostra en la Figura 2. Açò és important per a, posteriorment, comprendre com generar una funció utilitzant descodificadors binaris amb eixides actives a nivell alt.

Equació 1 Expressions algebraiques per les eixides d'un descodificador binari de 2 a 4 amb eixides actives a nivell alt

$$S0 = \bar{B} \cdot \bar{A} = \sum_{B,A}(0)$$

S0 prendrà valor 1 si B = A = 0

$$S1 = \bar{B} \cdot A = \sum_{B,A}(1)$$

S1 prendrà valor 1 si B = 0 i A = 1

$$S2 = B \cdot \bar{A} = \sum_{B,A}(2)$$

S2 prendrà valor 1 si B = 1 i A = 0

$$S3 = B \cdot A = \sum_{B,A}(3)$$

S3 prendrà valor 1 si B = 1 i A = 1

4.2 Forma Canònica Disjuntiva

Una forma senzilla de construir una funció lògica és desenvolupar la forma canònica disjuntiva, coneguda també com a suma de productes o suma dels minitermes de la funció.

---

[2] El valor de les eixides és S0=0, S1=1, S2=0, S3=0, perquè BA=01 es correspon amb el valor binari 1, per la qual cossa s'activa l'eixida S1 amb valor 1 (alt).

[3] Un miniterme és el producte de totes les variables d'entrada, que apareixen en forma directa si el seu valor és 1 i en forma negada si el seu valor és 0.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

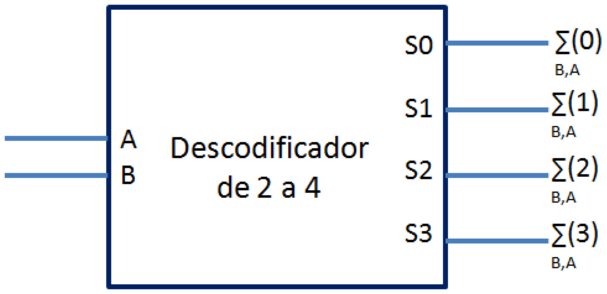

Figura 2 Descodificador 2 a 4 amb identificació dels minitermes

Quins són els minitermes d'una funció? Són aquells per als quals la funció pren valor 1. I la forma canònica disjuntiva diu que una funció és la suma dels seus minitermes. Però, millor ho veiem amb un exemple. La Taula 2 mostra una funció de nom G i aritat 3, i l'Equació 2 mostra la forma canònica disjuntiva d'aquesta funció.

Taula 2 Taula de veritat de la funció G

Miniterme	C B A	G
0	0 0 0	0
1	0 0 1	1
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

Equació 2 Forma Canònica Disjuntiva per a la funció G

$$G = \sum_{C,B,A}(1,2,4,6,7)$$

Per a construir el circuit que implementa la funció G es pot utilitzar portes lògiques, implementat els minitermes de la funció amb portes AND, i utilitzant una porta OR per a sumar els minitermes. En total, comptant les portes NOT necessàries per a construir els minitermes, necessitem _______[4] portes.

4.3 Generació de funcions amb descodificadors

De la mateixa manera que abans, podem crear una funció seguint els passos mostrats en la Figura 3.

L'últim pas correspon a la implementació dels minitermes de la funció mitjançant portes NOT i portes AND, i a la suma utilitzant una porta OR. Però, com hem vist anteriorment, un descodificador binari amb eixides actives a nivell alt implementa, en cadascuna de les seues eixides, un miniterme, per la qual cosa la primera part de la construcció del circuit pot ser substituïda per un descodificador binari, amb el

[4] Són necessàries una porta OR de 5 entrades, 5 portes AND de 3 entrades, i 3 portes NOT. En total, 9 portes.

mateix nombre d'entrades que el nombre de variables d'entrada de la funció lògica que es vol implementar.

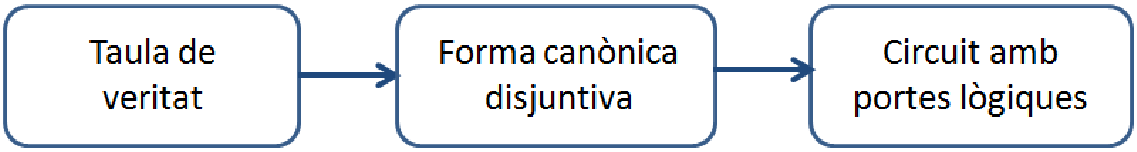

Figura 3 Passos per implementar una funció lògica

4.3.1 Implementació amb una porta OR

Podem aplicar la propietat associativa per a la suma, mostrada en l'Equació 3, a la suma de minitermes de la funció d'exemple G, mostrada en l'Equació 4.

Equació 3 Propietat associativa per a la suma

$$(a + b + \cdots + n) = a + b + \cdots + n$$

Equació 4 Aplicació de la propietat associativa a la forma canònica disjuntiva de la funció G de l'exemple de la Taula 1

$$G = \sum_{C,B,A}(1,2,4,6,7) = \sum_{C,B,A}(1) + \sum_{C,B,A}(2) + \sum_{C,B,A}(4) + \sum_{C,B,A}(6) + \sum_{C,B,A}(7)$$

Bé, el que ens diu la propietat associativa en aquest cas és que podem agafar els minitermes d'una funció, és a dir, les eixides corresponents del descodificador, i sumar-les mitjançant una porta OR. I l'eixida d'aquesta porta OR correspon amb la implementació de la funció. La Figura 4 mostra la implementació de la funció G utilitzant un descodificador de 3 a 8 amb eixides actives a nivell alt i una porta OR.

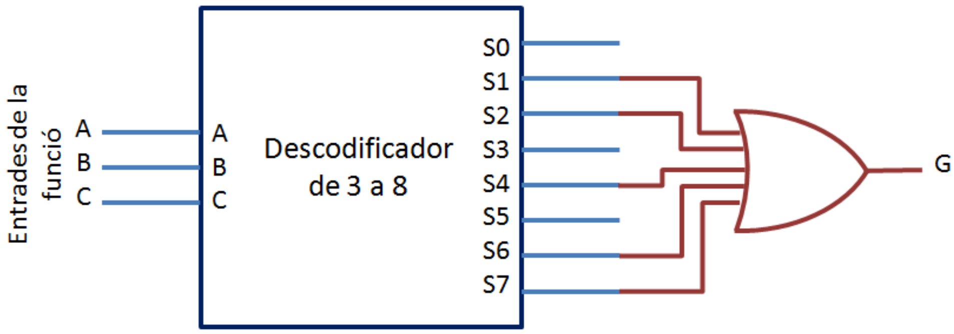

Figura 4 Implementació de la funció G per mitjà d'un descodificador binari i una porta OR

D'aquesta manera, per implementar una funció lògica només necessite un descodificador binari amb el mateix nombre d'entrades que l'aritat de la funció i una porta OR.

4.3.2 Implementació amb una porta NOR

Però, què passa si no puc aconseguir la porta OR que necessite?

Fem una altra pregunta. Què succeeix si en compte d'agafar els minitermes que **SÍ** que són de la funció, agafem els que **NO** són de la funció?

Açò seria el mateix que canviar els zeros per uns i els uns per zeros en l'eixida de la funció. Per exemple, amb la funció G, tindríem la taula de veritat mostrada en la Taula 3, amb la funció $\overline{G}$, la negació de la funció G. L'Equació 5 mostra la forma canònica disjuntiva per a $\overline{G}$.

Taula 3 Taula de veritat de les funcions G i $\overline{G}$

Miniterme	C B A	G	$\overline{G}$
0	0 0 0	0	1
1	0 0 1	1	0
2	0 1 0	1	0
3	0 1 1	0	1
4	1 0 0	1	0
5	1 0 1	0	1
6	1 1 0	1	0
7	1 1 1	1	0

Equació 2 Forma Canònica Disjuntiva per a la funció G

$$G = \sum_{C,B,A}(1,2,4,6,7)$$

Equació 5 Forma Canònica Disjuntiva per a la funció $\overline{G}$

$$\overline{G} = \sum_{C,B,A}(0,3,5)$$

Però nosaltres no volem implementar la funció $\overline{G}$, sinó que volem implementar la funció G. I açò ho podem aconseguir agafant els minitermes que **NO** són de la funció i utilitzar, en compte d'una porta **OR**, una porta **NOR**. D'aquesta manera aprofitem la propietat anomenada involució[5] i en negar $\overline{G}$ obtenim G, que és el que volíem.

La Figura 5 mostra la implementació de la funció G mitjançant l'ús d'un descodificador de 3 a 8 amb eixides actives a nivell alt i una porta NOR.

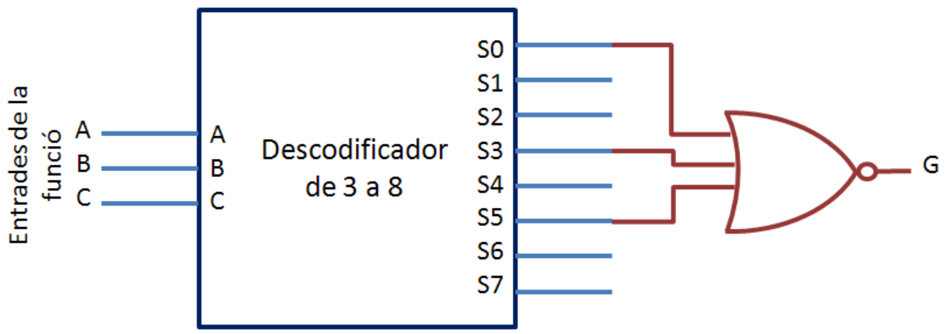

Figura 5 Implementació de la funció G utilitzant un descodificador binari i una porta NOR

[5] La propietat anomenada involució diu que $\bar{\bar{a}} = a$, es a dir, que si neguem una funció dues vegades, obtenim la funció original.

4.3.3 Resum

El resultat d'utilitzar una porta OR triant els minitermes que SÍ que són de la funció és el mateix que el d'utilitzar una porta NOR triant els minitermes que NO són de la funció. Així, doncs, a nivell funcional les dues opcions són equivalents. Tanmateix això, algunes funcions tenen un nombre més gran d'uns que de zeros en les seues eixides, o a l'inrevés. En aquests casos, la utilització d'una porta OR o d'una porta NOR pot tenir importància en la complexitat i cost del circuit final, ja que és possible triar la porta amb menor nombre d'entrades.

4.3.4 Exercicis

Seguidament uns exercicis senzills. Per a la funció lògica F expressada per la seua forma canònica disjuntiva, respon les qüestions següents:

$$F = \sum_{D,C,B,A} (1,2,6,7,11,14,15)$$

a. Quina grandària de descodificador binari necessitem per implementar-la?
b. Si utilitzem una porta OR, quantes entrades cal que tinga la porta?
c. Si utilitzem una porta OR, quines seran les eixides del descodificador que connectarem a la porta OR?
d. Si utilitzem una porta NOR, quantes entrades cal que tinga la porta?
e. Si utilitzem una porta NOR, quines seran les eixides del descodificador que connectarem a la porta NOR?
f. És millor utilitzar una porta OR o una porta NOR?
g. Prova de respondre les preguntes abans de veure les solucions, per favor [6].

5 Conclusions

En aquest article has pogut conèixer una forma ràpida i senzilla d'implementar una funció lògica. La Figura 6 mostra els passos que cal seguir per a, a partir de la taula de veritat d'una funció lògica, arribar a la implementació de la funció utilitzant un descodificador amb eixides actives a nivell alt i una porta OR o una porta NOR.

Algunes idees importants que cal recordar:

- El descodificador ha de tenir tantes entrades com a entrades tinga la funció. És a dir, ha de tenir la mateixa aritat que la funció.
- L'entrada de menor pes de la funció ha de connectar-se a l'entrada de menor pes del descodificador. I així successivament amb les entrades següents fins a la de major pes.
- Les eixides del descodificador que no s'utilitzen es deixen a l'aire.
- No és millor utilitzar una porta OR o una porta NOR, però depenent de la funció, és possible que resulte més senzill una opció enfront de l'altra. Però funcionalment les dues opcions són idèntiques.
- Si el descodificador té entrada d'habilitació, aquesta ha d'estar sempre activada, ja que en cas contrari no es genera cap funció.

---

[6] a: De 4 entrades a 16 eixides; b: De 7 entrades; c: S1, S2, S6, S7, S11, S14, S15; d: De 16-7=9 entrades; e: S0, S3, S4, S5, S8, S9, S10, S12, S13; f: La porta OR necessita dues entrades menys que la porta NOR, por la qual cossa pareix millor gastar la porta OR.

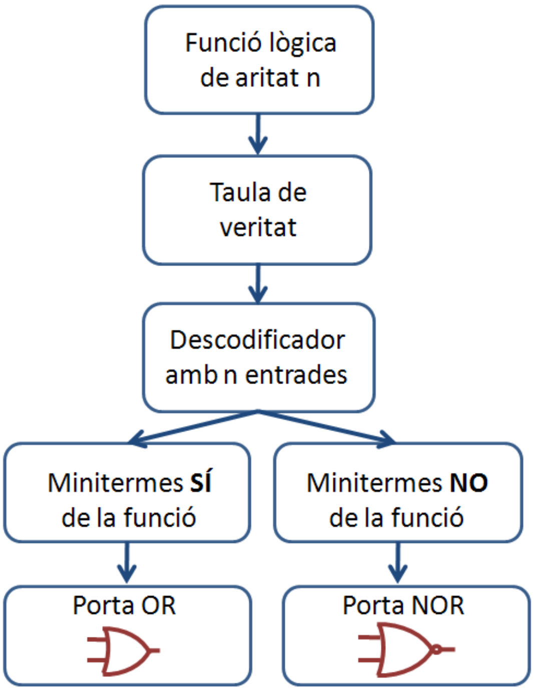

Figura 6 Passos per a la implementació d'una funció lògica utilitzant descodificadors binaris amb eixides actives a nivell alt

6 Bibliografia

6.1 Llibres:

[1] John F. Wakerly *Digital design : principles and practices*, Upper Saddle River : Pearson Prentice Hall. 2006

[2] Antonio Lloris Ruiz; Alberto Prieto Espinosa; Luis Parrilla Roure *Sistemas digitales*, Aravaca, Madrid : McGraw-Hill/Interamericana de España. 2003

6.2 Referències de fons electròniques:

[3] Martí Campoy, Antonio "*Circuitos combinacionales: decodificadores*" Universitat Politècnica de València. Escola Tècnica Superior d'Enginyeria Informàtica. 2011. http://politube.upv.es/play.php?vid=46040