

NOVES PERSPECTIVES DE DISSENY DE CIRCUITS AMB MEMÒRIES

J. AGUILÓ, E. VALDERRAMA, F. FABREGAT

Les tècniques d'integració actuals fan necessari mantenir al dia els algorismes fets servir al disseny. En aquest sentit l'objectiu d'aquest article és presentar un estat de la qüestió del tema de la descomposició modular de màquines seqüencials fent servir o bé mòduls més senzills connectats de forma serial-paral·lela o bé mòduls donats previamente (registres de desplaçament, ULMs, etc). Finalment s'estudia també la síntesi amb elements de memòria RAM.

1. INTRODUCCIÓ

Els mètodes clàssics de síntesi de circuits seqüencials presenten, a la pràctica, una sèrie d'inconvenients que el dissenyador no està en disposició de sortejar. La seva complexitat de tractament és gairebé infinita enfront de la seva eficàcia operativa. És per això que les noves perspectives apunten cap a una "standardització" de les funcions ja que això contribuirà a l'abaratiment del producte, a la utilització de mòduls LSI cada vegada més complexos i a la implementació amb memòries d'alta densitat cada vegada més petites i més barates. O bé, amb un altre punt de mira, que no tractarem, es tempta augmentar la densitat d'informació per connectar a base d'implementar lògiques de més de dos nivells. Dissortadament els problemes tecnològics que comporta aquesta última opció encara no estan resolts.

En aquest paper tractarem de passar revista en primer lloc al tema genèric de la descomposició de màquines seqüencials (m.s.), buscant els seus fonaments a teories fetes per resoldre el problema de la codificació d'estats interns. En segon lloc tractarem el problema de la descomposició funcional en els casos en què, o bé ens interessa per si mateixa, o bé perquè ens ofereix la possibilitat d'emprar un mòdul funcional donat. Per últim, donarem un cop d'ull a la síntesi amb RAM's gairebé només per advertir que té exactament els mateixos problemes que la res-

ta de tractaments.

La bibliografia que citem al final de l'article no és pas necessàriament mencionada en el text. Són només referències a llibres o articles de revista que els autors creuen d'interès per al lector que interessat una mica pel problema -desitjaríem que motivat en alguna mesura per aquest paper- tingui ganas d'aprofundir una mica en qualsevol dels temes.

2. CODATGE D'ESTATS

El problema, no resolt encara, de la codificació d'estats interns a una màquina seqüencial representa el punt de partida de la descomposició modular.

2.1 Si $Q = \{q_1, q_2, \dots, q_n\}$ representa el conjunt d'estats interns d'una màquina seqüencial M , $\Sigma \subset B^m$ el conjunt de símbols d'entrada i δ la funció d'estat següent $\delta : Q \times \Sigma \rightarrow Q$; una codificació d'estats no és més que una aplicació $\sigma : B^n \rightarrow Q$ que actuant sobre la funció d'estat següent indueix una nova aplicació, δ' , que pot ésser sintetitzada amb els elements de memòria discrets convencionals.

$$\begin{array}{ccc} \delta : Q & \times \Sigma & \longrightarrow Q \\ \sigma^{-1} \downarrow & & \downarrow \sigma^{-1} \\ \delta' : B^n & \times \Sigma & \longrightarrow B^n \end{array}$$

- J. Aguiló, E. Valderrama, F. Fabregat. Dept. d'Informàtica de la Universitat Autònoma de Barcelona. Bellaterra (Barcelona).
- Article rebut el Setembre del 1980.

2.1.1 Màquines autònomes

En el cas particular de màquines autònomes, $\Sigma = \emptyset$, la composició anterior és de la forma:

$$\delta' : B^n \rightarrow Q \rightarrow Q \rightarrow B^n \quad \delta' = \sigma \delta \sigma^{-1}$$

on, per construcció, δ i δ' són elements conjugats, és a dir, tenen grafs associats isomorfs.

Si interpretem δ' com un moviment del n-cub booleà B^n , es pot veure que /1/

δ' és una isometria del n-cub \Leftrightarrow la màquina seqüencial pot ésser dissenyada a ffd exclusivament.

De manera que, qualsevol funció d'estat següent, δ , del grup simètric de 2^n elements, d'una classe de conjugació, Δ , que contingui alguna isometria podrà ésser sintetitzada exclusivament amb elements de memòria, D, RS, JK ó GL.

D'altra part, existeix /36/ una correspondència entre les m.s. de 2^n estats sintetitzables sense lògica combinacional i les matrius $n \times n$ amb un element no nul per fila (1 ó -1) que ens permet una classificació ràpida de les m.s. i, per tant, la identificació del codi òptim en aquest cas.

2.1.2 Màquines no autònomes

Si considerem que una m.s. amb m entrades externes és un conjunt de 2^m màquines independents sobre els mateixos estats interns, és evident a partir dels raonaments anteriors que sempre que existeixi un codi òptim comú a cada una de les 2^m m. autònomes es pot aconseguir un disseny en la forma general -- com mostra la figura 1.

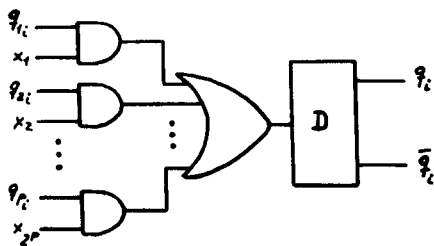


Fig. 1

amb un cost (segons la definició habitual -- /37/) igual o més petit que

$$n \cdot (2^{m+1} \cdot (m+1) - 2)$$

si s'utilitzen portes AND-OR de dues entrades.

$$2^{m+1} + n(2^{m+2} - 2)$$

si tot utilitzant portes AND-OR de dues entrades es minimitza com a funció múltiple.

$$n \cdot (2^m(m+2))$$

si s'utilitzen dos nivells de portes.

El mètode esmentat ens permet, doncs, una "standardització" del disseny en el sentit que el circuit combinacional d'entrada als elements de memòria és un multiplexador de 2^{m+1} entrades.

Els resultats a què arriba la descomposició polilineal a través de particions connexes i fortament connexes sobre el conjunt d'estats són qualitativament semblants als obtinguts encara que en determinats casos aconseguim funcions de cost millors a partir també d'un multiplexador "standard".

2.2 Codatge i particions

Si en el conjunt Q d'estats interns d'una m.s. definim una certa partició, π , designarem com a $PI(\pi)$ o partició induïda per π aquella que té per elements el conjunt d'estats següents associats a cada una de les classes de π . Direm que una partició π té la propietat de substitució, o que π és una p.p.s., si la seva partició induïda coincideix amb ella mateixa.

Per una m.s. donada, les p.p.s. formen un reticle característic. En particular per m.s. autònomes i completes amb les operacions de suma i producte les p.p.s. formen un anell.

Donada una m.s. de R estats, $2^n \geq R > 2^{n-1}$, i un nombre r de particions tals que

$$\pi_1 \cdot \pi_2 \cdot \dots \cdot \pi_r = 0$$

podem associar a cada un dels elements d'una partició π_i un cert nombre de variables in-

ternes K de manera que si M és el nombre màxim d'estats que conté un element de π_i , sigui $2^{K_i} M > 2^{K-1}$. En aquestes condicions, el nombre total S de variables internes necessàries per donar un codi associat a les r particions serà

$$S = \sum_{i=1}^r S_i$$

on S_i és el nombre de variables associat a la partició π_i . En general serà $S \gg n$ i la codificació no serà, òbviament, òptima.

Si $K(\pi)$ és el nombre d'elements d'una partició π i $L(\pi)$ el nombre d'estats de la classe més nombrosa i prenem

$$K = \lceil \log_2 K(\pi) \rceil, L = \lceil \log_2 L(\pi) \rceil$$

on $\lceil x \rceil$ indica part entera per excés, és clar que necessitarem K variables per codificar a cada una de les classes i L variables per diferenciar els estats dins de cada classe. Perquè la codificació sigui òptima s'ha d'imposar que $n = K + L$.

2.2.1 Exemple

Sigui la m.s. donada per a la taula d'estats següent

	\bar{x}	x	
1	3	7	
2	4	8	
3	1	6	
4	2	5	
5	1	4	
6	2	3	
7	4	1	
8	3	2	

i prenem, per exemple, les tres p.p.s. següents

$$\pi_1 = \{\overline{1234}, \overline{5678}\}; \pi_2 = \{\overline{1278}, \overline{3456}\};$$

$$\pi_3 = \{\overline{1468}, \overline{2357}\}$$

que compleixen que $\pi_1 \cdot \pi_2 \cdot \pi_3 = 0$, que $K_i + L_i = 3 \forall i$ i que $K_{ij} + L_{ij} = 3$ per a tots els productes $\pi_i \cdot \pi_j$.

Si associem tres variables internes y_1, y_2, y_3

a cada una de les particions π_1, π_2 i π_3 respectivament, els mapes de transició per a cada una d'elles seran

y_1	\bar{x}	x	y_2	\bar{x}	x	y_3	\bar{x}	x
0	0	1	0	1	0	0	1	1
1	0	0	1	0	1	1	0	0

de manera que

$$D_A = X \bar{A} \quad D_B = \bar{X} \oplus B \quad D_C = \bar{C}$$

3. DESCOMPOSICIÓ DE MÀQUINES SEQUËNCIALS

Direm que un conjunt, $\{M_1, \dots, M_r\}$ de r m.s. és una descomposició de M si el conjunt depèn de les mateixes entrades externes i el seu graf de comportament és isomorf al de M .

Realment cada subsistema M_i realitza "una part" de l'estat total de M . En general si δ_i és la funció d'estat següent per a la m.s., M_i serà:

$$\delta_i = f(Q_i, \Sigma, I)$$

és a dir, dependrà del propi estat intern, de les variables externes a M , Σ , i de les variables externes al sistema M_i però internes a M , és a dir, de la informació que li trameta la resta de subsistemes.

Direm que el subsistema M_j precedeix el M_i si M_j trameta informació a M_i . Direm que un conjunt de m.s. $C(M_i)$ és tancat si conté tots els seus predecessors.

3.1 Teorema fonamental de la descomposició

Si $\{M_1, \dots, M_r\}$ és una descomposició de M , a cada conjunt tancat $C(M_i)$ li correspon una p.p.s., π_i . Al conjunt de tots els $C(M_i)$ de la descomposició de M li correspon un conjunt de particions tals que

$$\prod \pi_i = 0$$

és a dir, el conjunt de particions associades als $C(M_i)$ constitueixen una codificació completa de M .

Si definim una relació d'ordre parcial sobre el conjunt de les particions es pot veure que /16/ la relació de precedència de subsistemes

és equivalent a la d'ordre de les particions. Al contrari, si π_i i π_j no són comparables - donaran lloc a conjunts de variables no dependents o amb altres paraules, a sistemes no lligats a cap relació de precedència. Parlarem en aquest cas de composició "paral·lel" (veure Figura 2.a) i en el anterior de composició "sèrie" (veure Figura 2.b).

3.1.1 Exemple

Considerarem la m.s. donada per la taula següent

	z			
	\bar{x}	x	\bar{x}	x
1	5	4	0	0
2	7	1	0	0
3	3	6	0	1
4	6	3	0	0
5	1	7	0	0
6	4	5	0	0
7	2	1	1	0

Podem veure sense dificultat que

$\pi_1 = \{\overline{135}, \overline{2467}\}$ és una p.p.s. i que $\pi_2 = \{\overline{16}, \overline{45}, \overline{23}, \overline{7}\}$ no és una p.p.s. però $\pi_1 \cdot \pi_2 = 0$.

La primera partició té dos elements, A i B - respectivament i la segona quatre que anomenarem C, D, E i F.

La descomposició de M, seguint aquest criteri serà (veure Fig. 3).

Y ₁	Y ₂	Y ₃
A	C	1
A	D	5
A	E	3
A	F	-
B	C	6
B	D	4
B	E	2
B	F	7

on cada un dels subsistemes tindrà el comportament següent

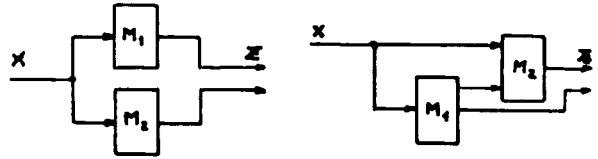


Fig. 2.a Composició paral·lel

Fig. 2.b Composició sèrie

M₁

	\bar{x}	x
A	A	B
B	B	A

M₂

	\bar{x}		x	
	A	B	A	B
C	D	D	D	D
D	C	C	F	E
E	E	F	C	C
F	-	E	-	C

i si assignem codis binaris a cada una de les variables, per exemple, A = 0, B = 1, C = 00, D = 01, E = 11, F = 10, la síntesi en biestables T serà:

$$T_A = X, T_B = XC + XAB, T_C = \bar{B}\bar{C} + \bar{X}\bar{B} + \bar{X}A + X\bar{A}C + ABC$$

$$Z = X\bar{A}BC + XAB\bar{C}$$

3.2 Cas general de descomposició

Sobre un cas qualsevol, amb una sèrie d'estats donat i un comportament també donat és molt probable que no existeixi cap p.p.s. (vegeu 6 per algorisme de recerca de p.p.s.) i que el mètode anterior no sigui directament aplicable. Això no vol dir pas que no existeixi una descomposició, sia sèrie o sia paral·lel.

Definirem una coberta amb propietat de substitució (c.p.s.) a un conjunt format per subconjunts d'estats amb les següents condicions

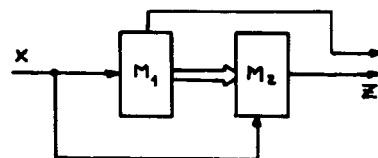


Fig. 3 Descomposició de M

1) cadascun dels estats pertany almenys a un dels subconjunts 2) tots els estats del mateix subconjunt tenen com a estats següents estats que són d'un mateix subconjunt (cal fer notar que aquests "subconjunts" no constitueixen una partició de Q. Un mateix estat pot estar, doncs, dins de diversos subconjunts).

Com en el cas anterior, l'existència d'una c.p.s. significa l'existència d'una descomposició sèrie amb un mòdul M_1 sense precedents lligat a la c.p.s., i un mòdul M_2 per diferenciar els estats dins d'un mateix subconjunt. També com en el cas anterior, l'existència de dos c.p.s. ortogonals ens permet assegurar la descomposició "paral·lel" de la m.s. M inicial.

La possibilitat d'utilització de c.p.s. per a la descomposició ens permet d'assegurar l'existència d'una descomposició en qualsevol cas, ja que tota m.s. de R estats té, com a mínim, R-2 c.p.s. no trivials que són les corresponents a totes les possibilitats de fer blocs d'un cert nombre d'estats més petit que R. Cal dir que en determinats casos (quan només existeixen les R-2 c.p.s. i les dues de trivials) la descomposició pot consistir en una identitat en sèrie amb la "mateixa" m.s. Aquest últim seria, per exemple, el cas de les permutacions d'ordre d'un nombre primer.

3.2.1 Exemple

Si sobre l'exemple 3.2 agafem com a segona partició $\pi_2 = \{\overline{16}, \overline{45}, \overline{23}, \overline{37}\}$, π_1 i π_2 són c.p.s. i $\pi_1 \cdot \pi_2 = 0$ cosa que ens permetrà, doncs, una descomposició en paral·lel. Serà:

M_1	\bar{x}	x
A	A	B
B	B	A

M_2	\bar{x}	x
C	D	D
D	C	F
E	F	C
F	E	C

i, en definitiva, per a la mateixa codificació d'estats que l'emprada abans, serà:

$$T_A = X, T_B = (B+C)X, T_C = \bar{X} + (\bar{B}+C)X$$

4. DESCOMPOSICIÓ EN MÒDULS DONATS

Tractarem tot seguit el cas on s'imposa un disseny amb un mòdul funcional donat. Això representa un cas típic de disseny amb mòduls funcionals integrats.

4.1 Síntesi amb registres de desplaçament

Si una m.s. M és tal que la seva sortida depèn solament d'un nombre finit d'entrades successives es podrà reduir, òbviament, a un registre de desplaçament i un circuit combinatorial de sortida. En cas contrari, caldrà la utilització del registre de desplaçament juntament amb un cert mòdul amb memòria com a part seqüencial, i, com sempre, un circuit combinatorial per a donar la funció de sortida.

En qualsevol cas, distingir un o altre comportament dins de la taula de comportament no és trivial.

Donada una taula de comportament, definirem com a arbre d'ambigüitat aquell que té com a vèrtex el conjunt d'estats i els successors més pròxims a un nus format estan donats pel conjunt d'estats a què s'arriba des dels que componen el nus inicial per a una determinada entrada.

Si després d'un nombre finit n de nivells s'arriba al fet que tots els nusos contenen un sol estat, el sistema és realitzable amb un registre amb decalatge de n bits i un circuit combinatorial de sortida que s'obté fàcilment si se li fa correspondre a cada un dels estats un multicodi segons la successió d'entrades necessàries per a arribar a ell des del vèrtex de l'arbre d'ambigüitat.

Si l'arbre d'ambigüitat no arriba a una situació com la descrita després d'un nombre finit de nivells però es compleix que:

a) A un determinat nivell hi ha un nus amb un conjunt d'estats {E} que se succeeix a si mateix al cap d'un nombre finit de passos.

b) A un nivell per al qual {E} ja ha aparegut almenys un cop, no hi ha cap nus amb un nombre d'estats superior al de {E}.

En aquestes condicions és possible de fer una descomposició amb un registre de desplaçament de longitud igual al nombre de ni-

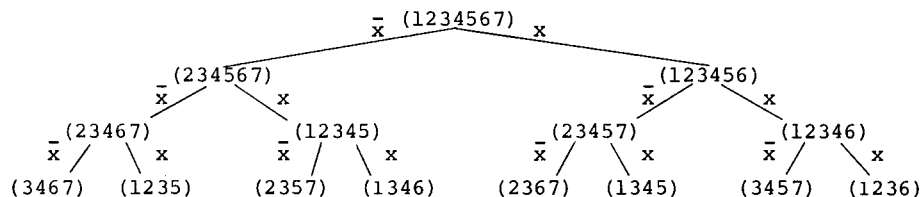
vells entre el vèrtex i la primera vegada - que es dona {E} i un circuit seqüencial amb nombre d'estats que serà, com a mínim, $\# \{E\}$ i que haurà de diferenciar entre els estats de cada un dels nusos. Com sempre, per a -- elaborar la funció de sortida ens caldrà un circuit combinacional.

4.1.1 Exemple

Sigui la taula d'estats següent

	\bar{x}	x	\bar{x}	x
1	5	6	0	0
2	3	3	0	0
3	7	1	0	0
4	3	3	0	1
5	2	4	0	0
6	4	2	0	0
7	6	5	1	0

El seu arbre d'ambigüitat serà:



El lector pot comprovar fàcilment que el nus (3467) es repeteix immediatament i que sobre el cinquè nivell (no posat a la figura) de l'arbre no hi ha cap nus amb més de quatre estats. Necessitarem, doncs, un registre de tres bits, un circuit seqüencial de quatre estats que determinarem i, com sempre, el circuit combinacional de sortida.

Per a determinar el circuit seqüencial podem assignar un codi fictici a cada un dels estats que apareixen a un nus i repetir el codi per a cada nus. Prenem, per exemple, el codi de la Fig. 4.

Aquest codi A, B, C i D donat per a cada nus d'acord amb el camí seguit des del vèrtex -- ens donarà la taula d'estats del mòdul seqüen

x_1	x_2	x_3	000	001	010	011	100	101	110	111
A			3	3	3	3	3	3	3	3
B			7	1	7	1	7	1	7	1
C			6	5	5	6	6	5	5	6
D			4	2	2	4	2	4	4	2

Fig. 4

cial que mostra la Fig. 5.

L'element D (marcat a la figura) s'obté de la manera següent; el codi C per al nus que correspon a les entrades 101 correspon, segons la taula de codis, a l'estat 5. Segons la taula d'estats inicial, per a una entrada \bar{x} el 5 ha de passar al 2, mentre que per a aquesta entrada el registre de desplaçament haurà esdevingut 010. Hem de buscar el codi corresponent a 2 per a la columna 010, que resulta que és efectivament D.

Encara que el procediment il·lustrat és gene

ral, per a aquest cas veiem que l'estat següent per al mòdul seqüencial és independent de l'estat del registre. Es tractarà de sintetitzar un mòdul autònom que respongui a la taula d'estats

A	B	0	2
B	C	2	3
C	D	3	1
D	A	1	0

El dispositiu (veure Fig. 6) ha de donar sortida 1 per a l'estat 7 amb entrada \bar{x} i per a l'estat 4 amb entrada x, és a dir, per unes posicions del registre (d d 0) (d, indica redundància) un codi B, (10) i entrada \bar{x} i per entrada x, codi D (01) i posicions del regis

	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111
A	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
B	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
C	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
D	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

Fig. 5

tre (000), (011), (101) i (110).

La funció de sortida serà, doncs:

$$f(x_1x_2x_3, y_1, y_2, x) = \bar{x}_3y_1\bar{y}_2\bar{x} + (\bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3) \bar{y}_1y_2x$$

4.2 Utilització de mòduls lògics universals

Considerem que un mòdul lògic universal de n-variables, ULM-n és un circuit combinacional que implementa la funció

$$f = \sum_{i=0}^{2^n-1} C_i(i_2, *, x)$$

seguit d'un retard; x representa el vector d'entrada, Σ la suma lògica, $C_i \in \{0,1\}$, * és l'operador potenciació, i_2 , és la representació binària de i.

Sobre la taula de comportament ens construïm la taula de precedències que tindrà sobre l'eix vertical els estats possibles de la m.s.; sobre l'horitzontal els símbols d'entrada; l'intersecció representarà els estats des dels quals s'arriba al de la fila per l'entrada a la qual correspon la columna. Encara que no l'utilitzem de moment, afegim una columna més, C_b , a la taula de precedència que ens indiqui el nombre total d'estats que apareixen a la fila.

Es pot veure /2/ que es pot fer la síntesi d'una m.s. qualsevol de R estats i n varia-

bles d'entrada amb un nombre R de ULM-n, un per a cada estat, solament prenent com a entrades les corresponents segons la taula de precedència.

4.2.1 Exemple

Sigui la taula d'estats i la de precedència corresponent les que s'expressen a continuació

	0	1	2	3
A	C	A	B	D
B	A	E	B	D
C	C	D	E	C
D	E	A	B	D
E	E	D	E	C

	I_0	I_1	I_2	I_3	C_b
A	B	AD	-	-	3
B	-	-	AB	D	3
C	AC	-	-	CE	4
D	-	CE	-	ABD	5
E	DE	B	CE	-	5

Taula d'estats

Taula de precedència

Segons el que hem dit al paràgraf anterior - ens caldran 5 ULM-2. El circuit resultant serà el que hi ha a la Fig. 8.

4.2.2 Utilització combinada de ULM i un decodificador

Modificacions /38/ del mètode de síntesi esmentat permeten, per utilització d'un decodificador, disminuir el nombre total de ULM's a canvi d'utilitzar més portes OR. Es tractarà de codificar els estats prenent compte -- que el nombre total de connexions ens ve donat per $N = \sum_{i=1}^R C_i x_i$ (nº de 1 en el codi de i) seran doncs interessants els codis que donin un nombre més petit de 1's als valors més grans de C_b . Les connexions a fer ens vin--

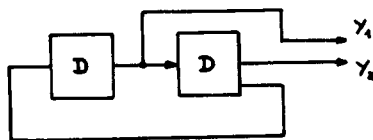


Fig. 6

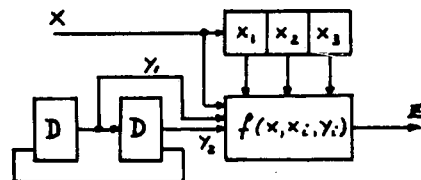


Fig. 7

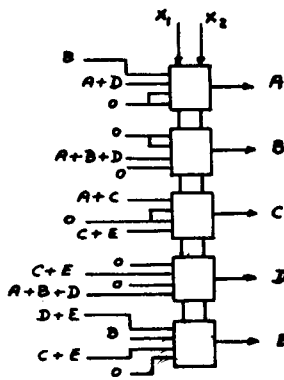


Fig. 8

dràn donades pels estats que figuren a la taula de precedència per a cada entrada com a corresponent al que té un 1 sobre la taula de codi.

4.2.3 Exemple

Sobre la mateixa taula de comportament anterior i tenint en compte el criteri donat, podem agafar el codi de tres bits següent:

	Y ₁	Y ₂	Y ₃
D	0	0	0
E	0	0	1
C	0	1	0
A	1	0	0
B	0	1	1

per a veure, per exemple, les connexions del segon mòdul hem de prendre sobre la taula de precedència les combinacions que corresponen a C i B. Així serà:

$$Y_2^1 = (A+C) I_0 + (C+E) I_3 + (A+B+D) I_2$$

Definitivament quedarà, després d'obtenir -- les altres dues equacions el esquema de la -- Fig. 9.

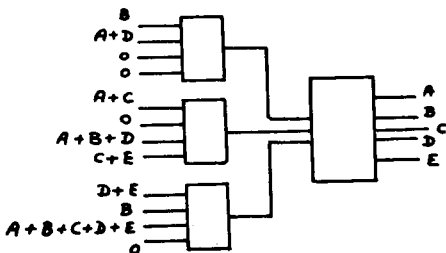


Fig. 9

4.3 Utilització d'un mòdul genèric

En aquest punt es pretén trobar una solució al problema de trobar amb quines condicions és possible, donat un mòdul conegut, (que serà, en general, un L.S.I.) i donada una m.s. trobar un procediment de síntesi o de descomposició segons el mòdul donat. Fins avui no s'ha arribat ni tan sols a fer una formulació assequible del problema. A pesar de tot, en determinats casos l'aplicació de l'algorisme que es dóna a continuació pot arribar a representar una solució no dolenta al problema -- plantejat.

Sigui M la màquina a sintetitzar, M' el mòdul LSI donat, δ i δ' les funcions d'estat següent respectives i q i q' dos estats genèrics. A partir de dos estats presos com a inicials -- q_0 i q'_0 ens construïm una "taula de transicions" de la següent manera:

$$\Delta_{\delta\delta'}: (Q \times Q') \times \Sigma \longrightarrow (Q \times Q')$$

$$(q, q', a) \longrightarrow (\delta(q,a), \delta'(q'a))$$

amb $(q, q') = \Delta^K(q_0, q'_0)$, és a dir, sense fer totes les parelles possibles del producte -- cartesià sinó prenent aquelles que s'obtenen a partir de l'estat inicial. Si no surten -- tots els estats de M s'ha d'agafar un segon estat inicial.

Si determinats $q \in Q$ quedessin associats a determinats $q' \in Q'$, $\Delta_{\delta\delta'}$ ens ha induït una partició de Q associada a M'. Com sempre haurem de trobar una altra partició per a poder distingir entre elements de la mateixa classe. Si s'escau, es pot reiterar l'aplicació de -- l'algorisme per tal de descompondre altre cop la funció que resulti.

5. SÍNTESI AMB MEMÒRIES D'ALTA DENSITAT

Esquemàticament, el que es pretén a l'hora -- de sintetitzar m.s. amb elements de memòria RAM es emmagatzemar a la RAM la taula d'estats de la m.s. en comptes de construir l'estructura hardware segons les especificacions.

Bàsicament, el problema que presenta aquest tipus de realització és el mateix que en els altres casos: com fer la codificació d'estats de manera que ocupi el mínim lloc possible -- de memòria, i en un format capaç d'ésser llegit i reconegut pel dispositiu?

L'esquema de funcionament que es pretén és - el de la Fig. 10.

Una entrada externa X_0 ha de fer llegir el - contingut del registre "estat present" i a - partir d'aquest i de X_0 configurar les adre- ces d'estat següent i de sortida continguts a la RAM. Aquest estat següent ha de despla- çar l'anterior, contingut al registre "estat present".

Si es considera que una m.s. amb m entrades externes és la unió de 2^m m.s. sense entrades es pot organitzar la taula d'estats següents /sortides amb subtaules, de manera que una - determinada entrada seleccionarà la subtaula i l'adreça d'estat següent és, tan sols, fun- ció de l'estat present.

És clar que els termes redundants de la tau- la d'estats no cal implementar-los a la RAM, i per altra banda podem donar la mateixa -- adreça d'estat següent a dos estats q_i i q_j que responguin de la mateixa manera dins de subtaules diferents. Es tractarà doncs, com dèiem, de donar una codificació adequada i - preseleccionar per a cada símbol d'entrada - un subconjunt de variables segons el proble- ma específic que es presenti.

5.1 Exemple

Sigui la taula d'estats

	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4
1	3	-	4	2	A	-	A	A
2	1	6	-	2	A	A	-	A
3	3	-	4	5	A	-	A	A
4	-	5	3	2	-	A	B	A
5	-	4	4	-	-	A	B	-
6	5	-	3	-	A	-	A	-
7	5	6	3	-	B	A	A	-

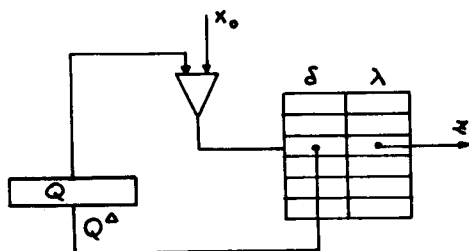


Fig. 10 Síntesi amb RAM

De l'observació d'aquesta resulta que per a la primera entrada x_1 només hi han 4 respos- tes diferents, 3 per a la segona, 4 per a - la tercera, i tan sols 2 per a l'última. Co- dicarem amb 4 variables cada un dels es- tats de manera que, a més, les combinacions de variables internes estiguin d'acord amb les variables d'entrada que ens han de defi- nir cada subtaula.

Així, si associem a x_4 la variable interna y_1 s'haurà de tenir en compte que la part - de codi a què correspon la variable y_1 ha - d'ésser el mateix per als estats 1, 2 i 4 i diferent per al tres. Això fa que no es - pugui emprar l'associació de $y_1 y_2$ amb la va- riabile x_1 ni amb la x_3 , ja que ambdues exi- geixen la igualtat de codis per als estats 1 i 3.

La codificació proposada /40/ per a aquest cas és la

	y_1	y_2	y_3	y_4
1	0	0	0	0
2	0	1	0	X
3	1	0	0	0
4	0	0	0	1
5	1	0	1	0
6	X	0	1	1
7	0	1	1	1

x_1	y_2	y_3
x_2	y_1	y_2
x_3	y_3	y_4
x_4	y_1	

que compleix amb totes les especificacions - exigides (prendrem com a 0 els elements redun- dants). Si ens ordenem la taula en subtaules que corresponguin a cada un dels símbols d'en- trada i per ordre creixent de codis d'estat present, ens queda

	δ	λ
x_1	1000	A
	1010	A
	0000	A
	1010	B
+ 4		
x_2	1010	A
	0011	A
+ 7	0001	A
	0001	A
x_3	0001	A
	1000	B
+ 11	0001	B
	1000	A
x_4	0100	A
	1010	A

Si el registre "estat present" conté, per -- exemple, 0011 i l'entrada és x_3 (associada a la posició +7), el pointer s'ha de situar a $(7+0011)_2=10$ que correspon a 1000 A o bé, des codificant, a l'estat 3, senyal de sortida A com es preveia.

La recerca es pot fer, a la pràctica, o bé - seguint una implementació hardware de l'algorisme o bé mitjançant un programa emmagatzemat en ROM.

6. CONCLUSIONS

Com ja dèiem al moment d'introduir el tema i com el lector deu haver pogut comprovar, el grau de complexitat a què ha arribat la tecnologia actual ha desbancat totalment les bases teòriques amb què, en el fons, es fonamenta.

Tractaments profunds i exhaustius de tots i cadascun dels temes que aquí hem plantejat - resten encara al tinter. Els projectes de recerca en aquests terrenys són perfectament - irrendables a curt termini; això es tradueix sense cap mena de dubte, en el fet que siguin solament les universitats les que s'encarreguin d'aquesta mena de feina ingrata que no permet gaudir dels resultats en un termini - breu de temps.

En qualsevol cas, el dissenyador no té prou eines per a decidir amb certesa quin algorisme ha d'emprar, i és l'experiència qui substitueix al raonament a la major part de casos. És possible, fins i tot, que només el plantejament del problema i la presa de decisió sigui un procediment més costós que no pas un disseny no òptim però fet amb un moment.

Malgrat tot això, hem volgut fer veure a través d'aquesta panoràmica sobre les "noves -- tendències" que en realitat els problemes -- d'avui són els mateixos que ahir. O és que - potser un disseny modular no és gaire semblant a una dependència reduïda que introdueix una codificació? O és que potser la problemàtica d'una implementació RAM no és la mateixa que està sense resoldre's des de 1943 i que ha estat atacada per tots els informàtics des de tots els punts de vista? Potser val la pena estudiar d'una vegada i per sempre els vells problemes per a poder suportar

els nous. Com deia molt recentment un manifest d'homes de ciència "s'han postergat els aspectes creatius de la recerca a un pragmatisme a curt termini propi d'una societat colonial", "la tecnologia ha d'estar en contacte directe i continu amb la investigació bàsica" ja que la primera no és del tot útil - si no es "recolza en un fort desenvolupament de la segona".

7. BIBLIOGRAFIA

- /1/ AGUILO, J.: "Circuitos secuenciales sintetizados exclusivamente con flip-flops" Tesis doctoral. Universidad Autónoma de Barcelona. Septiembre 1977.
- /2/ ALMAINI, A.E.A.: "Sequential Machine Implementations Using Universal Logic Modules". IEEE Trans. on Computers, Vol. c 27, nº 10, octubre 1978.
- /3/ ARMSTRONG, D.B.: "A Programmed Algorithm for Assigning Internal Codes to Sequential Machines". IRE Transactions on Electronic Computers. August 1962.
- /4/ CAVARROC, J.C.; BLANCHARD, M.; GILLON, J.: "An Approach to the Modular Design of Industrial Switching Systems". International Symposium on Discrete Systems. Riga, USSR. Octubre 1974.
- /5/ CIOFFI, G.; DE JULIO, S.; LUCERTINI, M.: "Optimal decomposition of sequential machines via integer non-linear programming: A Computational Algorithm". Digital Processes nº 5, 1979, pp. 27-41.
- /6/ COTTREZ, G.; BLANCHARD, M.; GILLON, J.; GUIDEZ, G.; THUILLIER, G.: "The Simulation of a Switching Systems Requirements" International Symposium on Discrete Systems. Riga, USSR, Octubre 1974.
- /7/ CHAO-CHIH YANG: "Structural Preserving - Morphisms of Finite Automata and an Application to Graph Isomorphism". IEEE Trans. on Comp., Nov. 1975.
- /8/ CHIN CHAO and HERSCHEL H. LLOMIS, Jr.: "High Rate Realization of Finite-State - Machines". IEEE Trans. on Comp. Vol. c24 nº 7, Juliol 1975.

- /9/ DAVID, R.: "Modular Design of Asynchronous Circuits Defined by Graphs". IEEE Trans. on Comp. Vol. c-26, n^o 8, Agost 1977.
- /10/ DAVIO, M. i BIOUL, G.: "Interconnection Structure of Injective Counters Composed Entirely of JK Flip-Flops". Information and Control. Vol. 33, n^o 4, Abril 1977.
- /11/ FRIEDMAN, A.D.: "Theory and Design of Switching Circuits". Pitman. London, - England, 1977.
- /12/ GAITANIS, N.; HALATSIS, C.: "A New Double-Rank Realization of Sequential Machines". IEEE Trans. on Comp. Vol. c27 n^o 12, Desembre 1978.
- /13/ GAITANIS, N.; HALATSIS, C. i SIGALA, M.: "Fail-safe commutators", Digital Processes n^o 5, 1979, pp. 43-57.
- /14/ GREENFIELD, J.D. "Practical Digital Design Using IC's". John Wiley and sons, 1977.
- /15/ HARING, D.R.: "Sequential Circuit Synthesis". Research Monograph n^o 31. The MIT Press. Cambridge, Massachussets, - 1966.
- /16/ HARTMANIS, J.: "Algebraic Structure -- Theory of Sequential Machines". Prentice Hall. 1966.
- /17/ HARRISON, H.J.; STORY, J.R.; REINHARD, F.A.: "Optimum State Assignment for - Synchronous Sequential Machines with - Flip-Flop Memory". Comunicació personal.
- /18/ HARRISON, M.A.: "Lectures on Linear Sequential Machines". Academic Press, -- 1969.
- /19/ HEBENSTREIT, J.: "Complements de Logique Sequentielle Synchrones. Decomposition Fonctionnelle". Cours d'Informàtica Teòrica. Institut de Programació de Paris.
- /20/ HILL, F.J.: "Teoría de conmutación y - diseño lógico". Limusa, 1978.
- /21/ KARTASHEV, S.P.: "State Assignment for Realizing Modular Input-Free Sequential Logical Networks Without Invertors". -- Journal of Computer and Systems Sciences, 7. 1973. pp. 522-542.
- /22/ KARTASHEV, S.P.; KARTASHEV, S.I.: "On - Modular Networks Satisfying the Shift-Register Rule". IEEE Trans. on Comp. -- Vol. c-27, n^o 12. Desembre 1978.
- /23/ KIDSON, D.: "A Consideration of the Use of Standard MSI units in the Realization of Sequential Machines". Digital - Processes, n^o 4, 1978. pp. 89-107.
- /24/ KING, W.F., III; WEINER, P.: "On the Modular Decomposition of Autonomus Sequential Machines". IEEE Computer Group, -- Switching and Automata Theory Committee, Univ. Waterloo, 1969.
- /25/ LEE, S.C.: "Digital Circuits and Logical Design". Prentice Hall. 1976.
- /26/ MAGO, G.: "Monotone Functions in Sequential Circuits". IEEE Trans. on Comp. V. c-22, n^o 10. Octubre 1973.
- /27/ MANGE, D.: "Synthese des compteurs synchrones a l'aide du flip-flop plus general". Agen. Desembre 1967.
- /28/ MANNING, F.B.: "An Approach to Highly - Integrated, Computer-Maintained Cellular Arrays". IEEE, Trans. on Comp. Vol. c-26 n^o 6. Juny 1977.
- /29/ MARCOVITZ, A.B.; PUGSLEY, J.H.: "An introduction to Switching System Design". Wiley Chichester, Sussex, England, 1971.
- /30/ PERRIN, J.P. et al.: "Systemes logiques" Tomo II. Dunod, 1967.
- /31/ ROOME, W.D.; TORNG, H.C.: "Algorithms - for Multiple Shift Register Realizations of Sequential Machines". IEEE Trans. on Comp. Vol. c-22, n^o 10. Octubre 1973.
- /32/ SAUER, A.: "Sequential System Structures". Siemens Forsch.- and Entwicklungsber (Alemania). Vol. 7 n^o 6. 1978. pp. 319-321.

- /33/ TIU LE VAN; VAN HOUTTE, N.: "Delayed - Universal Logic Modules and Sequential Machine Synthesis". IEEE Trans.on Comp. Agost, 1975.
- /34/ TORNG, H.C.; WILHELM, N.C.: "The optimal Interconnection of Circuit Modules in Microprocessor and Digital System - Design". IEEE Trans. on Comp. Vol. c26 n^o 5. Maig, 1977.
- /35/ ULUG, M.E.; BOWEN, B.A.: "A Unified -- Theory of the Algebraic Topological Methods for the Synthesis of Switching - Systems". IEEE Trans. on Comp. Vol. c-23, n^o 3, Març, 1974.
- /36/ VALDERRAMA, E.: "Asignación de estados en máquinas sin lógica combinacional". Tesi doctoral. Universitat Autònoma de Barcelona. Juliol, 1979.
- /37/ VALLS, J.: "Asignación de estados en - máquinas secuenciales no autónomas". - Tesina. Universitat Autònoma de Barcelona. Juny, 1979.
- /38/ VAN LANTSHOOT, E.J.: "Double Adjacen--cies Between Cycles of a Circulating - Shift Register". IEEE Trans. on Comp. Vol. c-22, n^o 10. Octubre 1973.
- /39/ VOITH, R.P.: "Minimum Universal Logic - Module Sequential Circuits With Deco--ders". IEEE Trans. on Comp. Vol. c-26. n^o 10. Octubre 1977.
- /40/ WILKENS, E.J.: "Realizations of Sequen--tial Machines Using Random Access Memo--ry". IEEE Trans. on Comp. Vol. c-27, - n^o 5, Maig, 1978.
- /41/ WILLIAMS, G.H.: "Uniform Decomposition of Incompletely Specified Sequential - Machines". IEEE Trans. on Comp. Agost 1975.