

Introducció

Aquest document mostra els canvis lògics que provocaria afegir 3 noves instruccions anomenades “LdD”, “AID”, “CmpD” i quins canvis físics s’haurien d’implementar.

Noves instruccions

L’enunciat demana implementar 2 noves instruccions (“AID” i “CmpD”) per a poder fer una operació aritmètica o una comparació fent ús del valor d’un registre juntament amb un valor de memòria.

Per la naturalesa de l’arquitectura, es requereix una tercera instrucció (“LdD”) que permeti guardar el valor de memòria, sinó el temps de cicle seria massa llarg.

Funcionament

Per complir els requisits, s’ha hagut d’implementar:

- 1 Biestable (“RMem”)
- 2 MUX-2-1 (“RY/RMem”, “Rb/R@-Pc”)

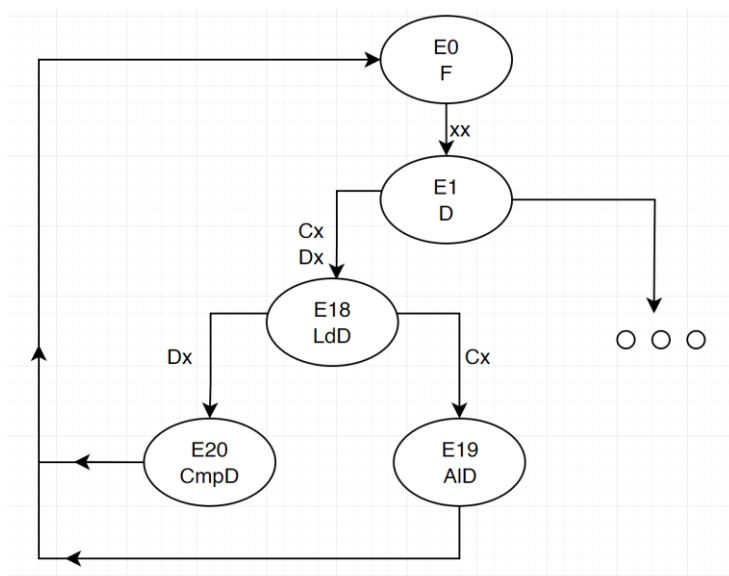
RMem: S’encarrega de guardar durant un cicle el valor que surt de la memòria.

RY/RMem: Serveix per decidir quin Biestable (“RY” o “RMem”) entra per l’entrada “Y” de l’ALU i poder fer l’operació/comparació.

Rb/R@-Pc: Permet decidir quin valor (“Rb” o “R@-Pc”) entrarà a la memòria. Això permet que el valor que surti per “B” del REGFILE pugi anar directe a l’entrada de la memòria i no perdre cicles guardant el valor en el registre “R@”.

Graf d’Estats

Les instruccions ja conegudes no canvien. S’afegeixen les 3 instruccions esmentades on “LdD” sempre s’efectuarà prèviament a “AID”/“CmpD” donat que es necessita guardar el valor que surt de la memòria.



Accions a Registres en cada Estat

Tant “Fetch” com “Decode” continuen igual.

“LdD” guarda en RMem el valor de memòria. És important entendre que això es possible donat que per accedir a aquest valor no s’ha de realitzar cap operació.

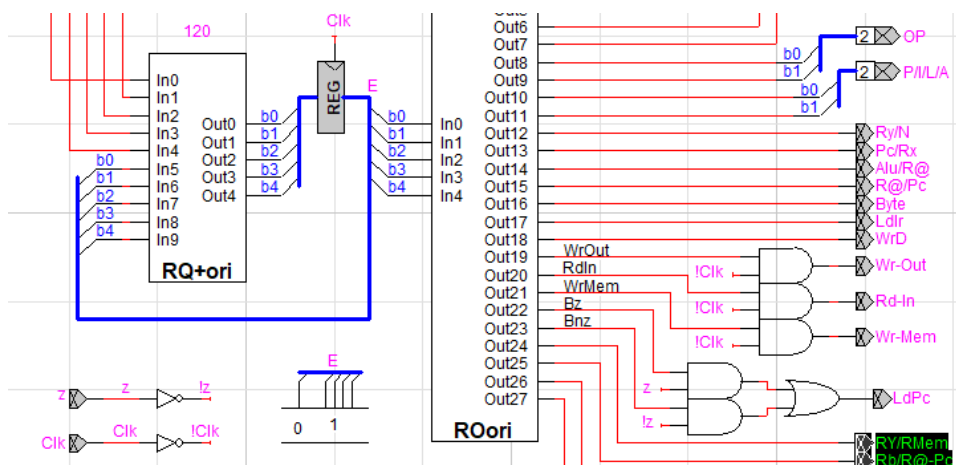
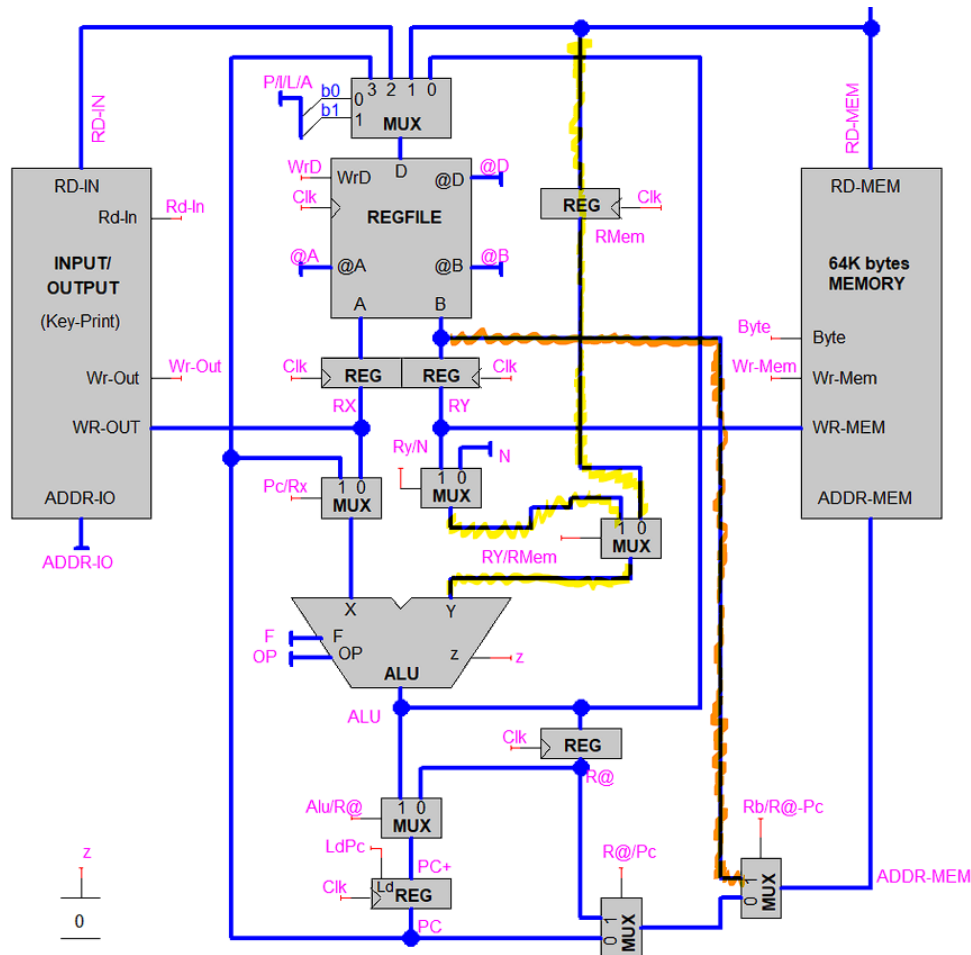
“AID” i “CmpD” fan la seva funció de forma transparent per l’ALU, només canvia quin valor de Biestable es fa servir.

Node/Estat		Accions
Número	Mnem.	(En llenguatge de transferència de registre)
E0	F	IR<-- MEMw[PC] // PC <-- PC+2
E1	D	R@ <-- PC+SE(N8)*2 // RX <-- Ra // RY <-- Rb
E18	LdD	RMem <-- MEMw[Rb]
E19	AID	Rd <-- RX AID RMem
E20	CmpD	Rd <-- RX CmpD RMem

Modificacions Hardware

S'afegeix 2 Multiplexors 2-1 per controlar quin Biestable entra per l'entrada Y de l'ALU, com quin valor entra per la memòria, si ho farà el valor de "B" o el de "R@/Pc"

És requereixen 2 noves senyals de control que les controlarà la ROM_OUT, es per això que s'hauran de modificar algunes instruccions per adaptar-les a aquesta nova arquitectura.



Les parts de color **gris** es mantenen igual. Les parts de color **groc** són les modificades.

[illegible]

Taula ROM_Q+

Donat que estem afegint noves instruccions, requereixen de nous estats en el graf d'estats. Es per això que haurem de modificar la ROM_Q+ que és l'encarregada de controlar en quin estat estem per a que pugi gestionar la nova arquitectura.

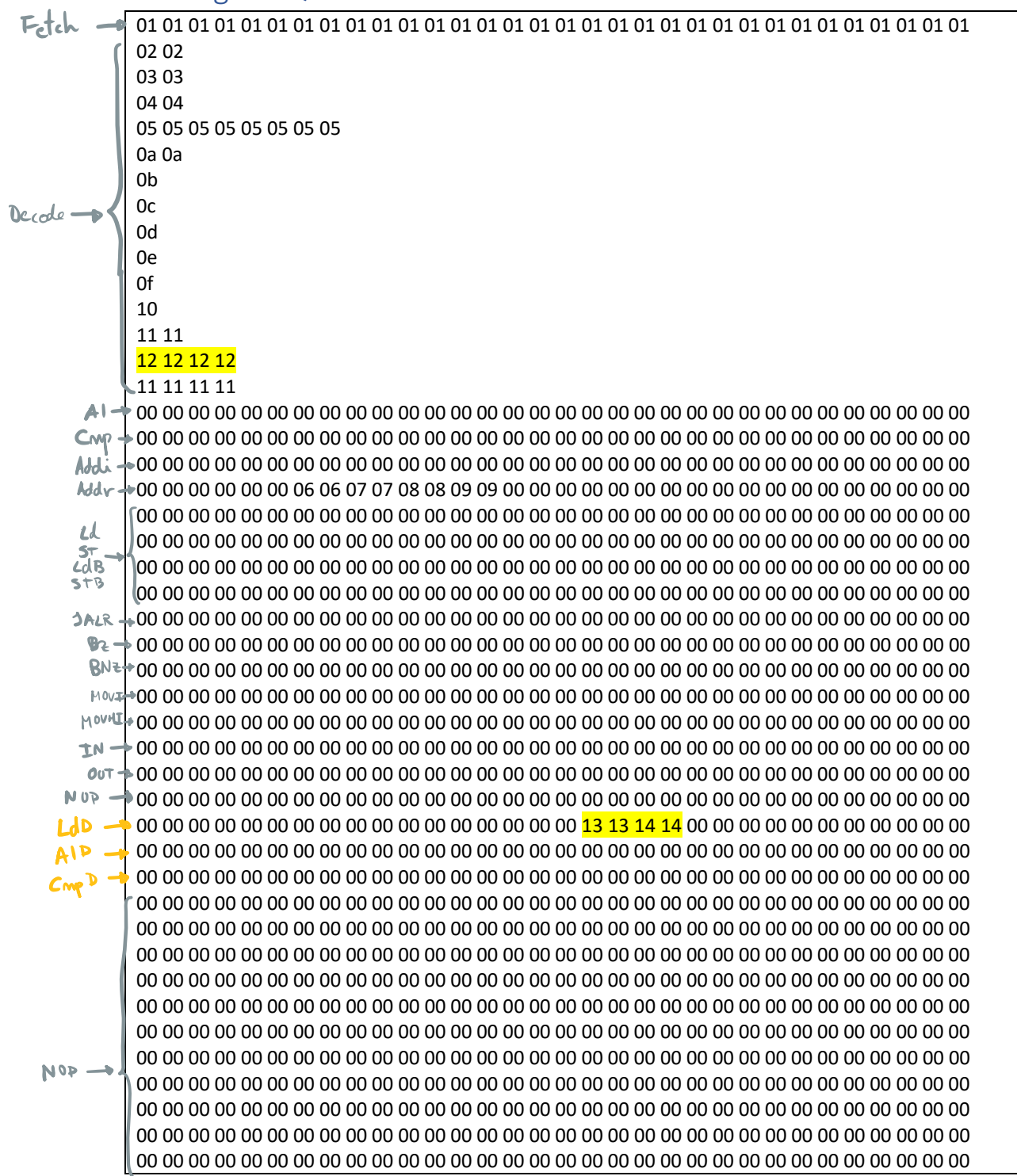
En aquesta taula podem veure les implantacions per les noves instruccions, la resta es manté igual.

Q	I	Q+
F	X	D
[...]		
D	AID	LdD
D	CmpD	LdD
[...]		
LdD	!(AID+CmpD)	x
LdD	AID	AID
LdD	CmpD	CmpD
[...]		
LdD	!(AID+CmpD)	x
AID	x	F
CmpD	x	F

q4	q3	q2	q1	q0	I15	I14	I13	I12	I8	Q+
0	0	0	0	0	x	x	x	x	x	"0x01"
[...]										
0	0	0	0	1	1	1	0	0	x	"0x12"
0	0	0	0	1	1	1	0	1	x	"0x12"
[...]										
1	0	0	1	0	0	x	x	x	x	x
1	0	0	1	0	1	0	0	x	x	
1	0	0	1	0	1	0	1	0	x	
1	0	0	1	0	1	0	1	1	x	"0x13"
1	0	0	1	0	1	1	0	0	x	"0x14"
[...]										
1	0	0	1	0	1	1	0	1	x	x
1	0	0	1	0	1	1	1	0	x	
1	0	0	1	0	1	1	1	1	x	
1	0	0	1	1	x	x	x	x	x	"0x00"
1	0	1	0	0	x	x	x	x	x	"0x00"

# Vegades	Q+
32	1
[...]	
2	"0x12"
2	"0x12"
[...]	
16	"0x00"
2	"0x00"
2	"0x00"
2	"0x13"
2	"0x14"
[...]	
2	"0x00"
2	"0x00"
2	"0x00"
32	"0x00"
32	"0x00"

Contingut RQ+ori



Les files representen en quin estat estem.

Les columnes representen a quin estat volem anar.

El valor que es veu és el valor en hexa del estat següent (Desde “Fetch” [E0] → “Decode” [E1])

Altres Implementacions

- En comptes de fer servir un nou registre, es podria haver ficat el multiplexor abans del registre “RY” i decidir si es guarda el valor de “B” o el que surt per memòria (També s’hauria d’haver afegit un cable que connectés la sortida de memòria amb aquest multiplexor).
- És va pensar si era possible fer servir l’estat “Al”/”Cmp” en comptes de crear-los. Això no és possible perquè donat la naturalesa de la meva implementació, requeria senyals addicionals i les instruccions ja implementades millor no tocar-les.