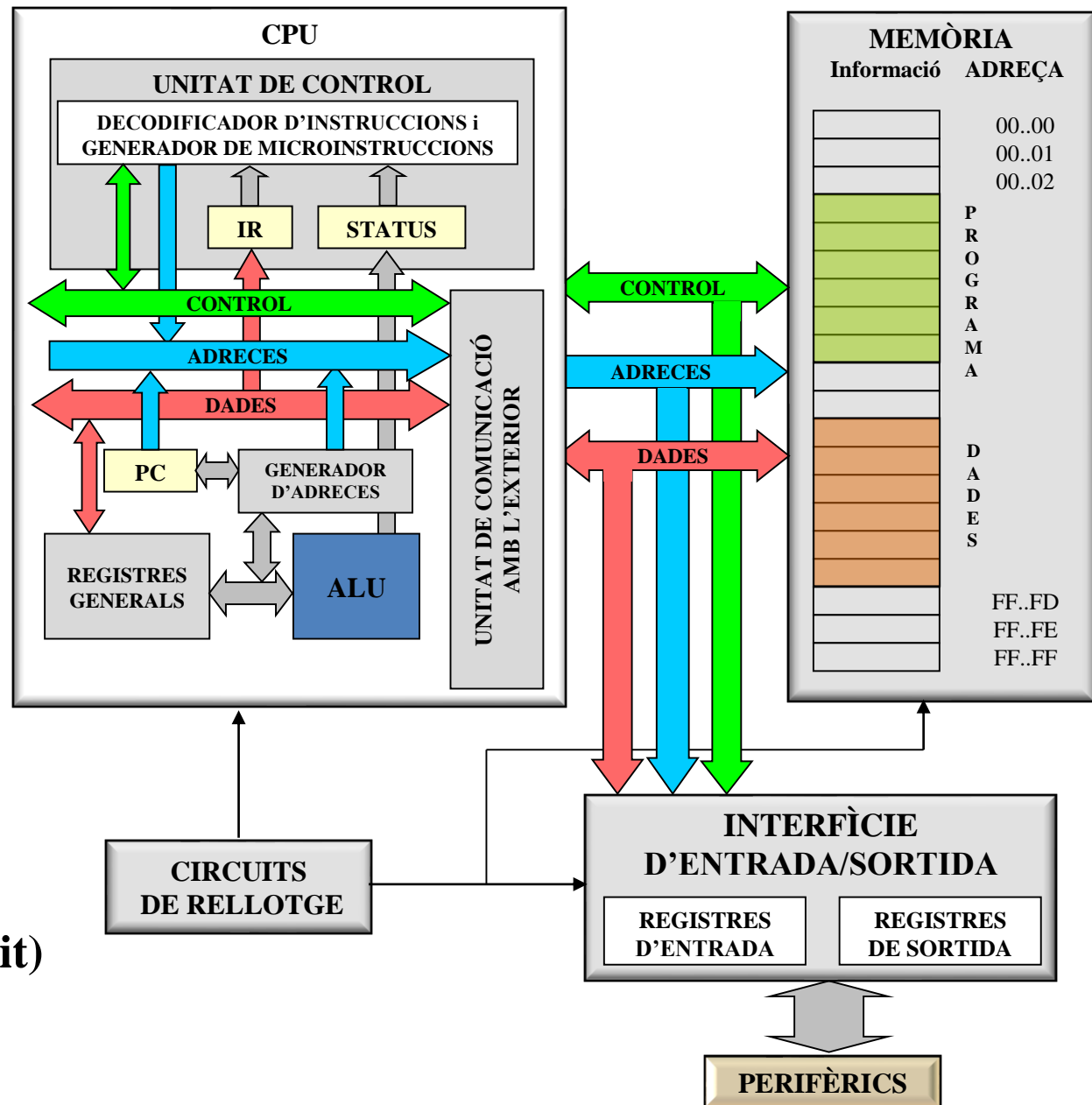


CPU (Central Process Unit)

Índex de conceptes

- **Unitat de Procés (ALU + Registre)**
- **Unitat de Control (comptador + ROM)**

Arquitectura (Von Neumann) d'un Processador



- Memòria Principal
- Dispositius d'E/S
- CPU (Central Process Unit)
 - ✓UC (Unitat de Control)
 - ✓UP (Unitat de Procés)

Implementarem una **CPU** senzilla que realitza la següent operació



$$W = \frac{2M + N - P + 1}{4}$$

on M, N i són quantitats de 4 bits. Aquesta CPU ha de tenir una **ALU** i un **registre (REG)**. Com que l'operació no es pot fer de cop, és necessari fer diverses operacions en el temps (fer un algoritme). Per tant cal un registre on desar el resultat de les operacions intermèdies i el resultat final. L'ordre de les operacions no està definit i hi ha diverses possibilitats, per exemple:

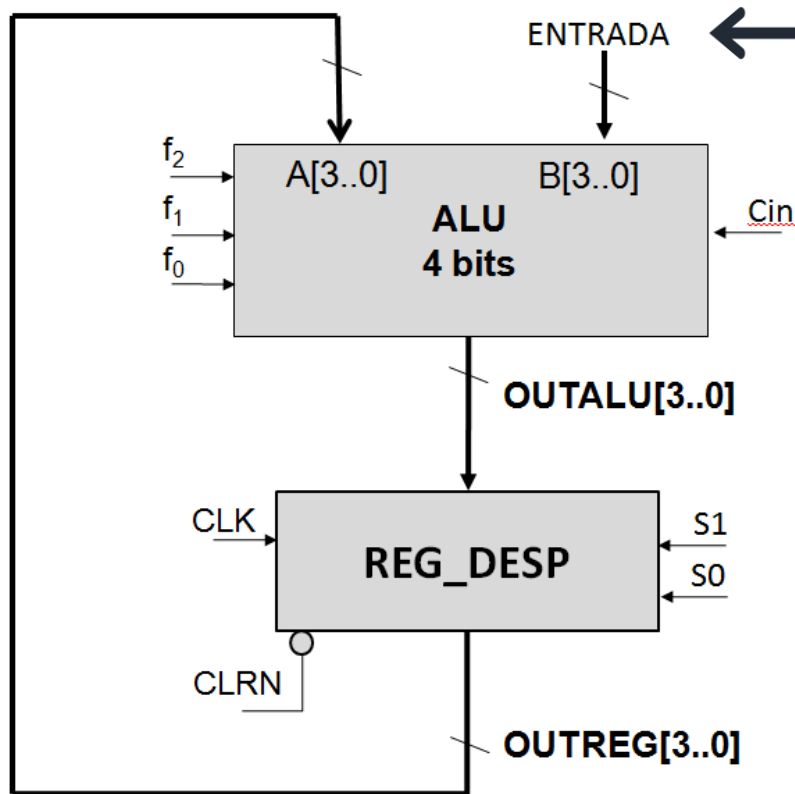
- Carregar l'operand M al registre
(M => REG)
- Multiplicar per 2 l'operand del registre i desar el resultat al registre
(2M => REG)
- Sumar l'operand N al contingut del registre i desar el resultat al registre
(2M+N => REG)
- Restar l'operand P al contingut del registre i i desar el resultat al registre
(2M+N-P => REG)
- Sumar 1 al contingut del registre i desar el resultat al registre
(2M+N-P+1 => REG)
- Dividir el contingut del registre per 2
(2M+N-P+1)/2 => REG)
- Dividir el contingut del registre per 2
(2M+N-P+1)/4 => REG)

Unitat de Procés

Implementarem una **CPU** senzilla que realitza la següent operació

$$\rightarrow W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.



Es disposa d'un senyal d'entrada de 4 bits per poder introduir els operands M, N i P de manera seqüencial

Unitat de Procés

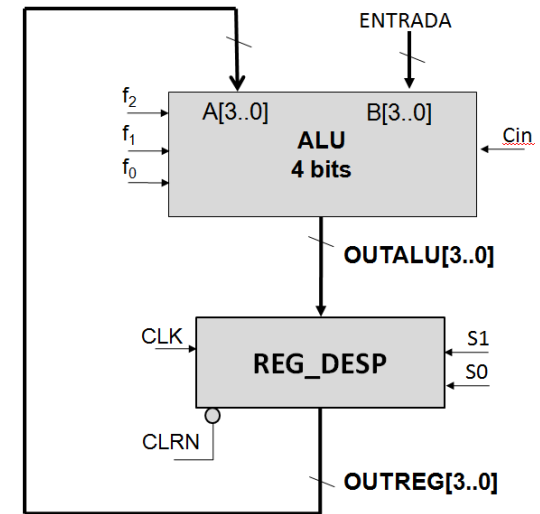
Implementarem una **CPU** senzilla que realitza la següent operació

$$\rightarrow W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.

Aquesta taula conté una llista de tots els senyals de la UP.

Nom	Descripció	E/S	Longitud
ENTRADA	Operand d'entrada.	Entrada	4b
ENTRADA	Cin d'entrada.	Entrada	1b
f2	Senyal de control d'operacions de l'ALU	Entrada	1b
f1	Senyal de control d'operacions de l'ALU	Entrada	1b
f0	Senyal de control d'operacions de l'ALU	Entrada	1b
CLK	Senyal de rellotge	Entrada	1b
CLRN	Senyal de <i>clear</i> . Actiu per nivell baix. Posa a zero el REG.	Entrada	1b
S1	Senyal de control del registre de desplaçament.	Entrada	1b
S0	Senyal de control del registre de desplaçament.	Entrada	1b
OUTALU	Contingut del ALU.	Sortida	4b
OUTREG	Contingut del registre. Conté el resultat final i permet monitoritzar l'execució de la UP.	Sortida	4b



Unitat de Procés

Implementarem una **CPU** senzilla que realitza la següent operació →

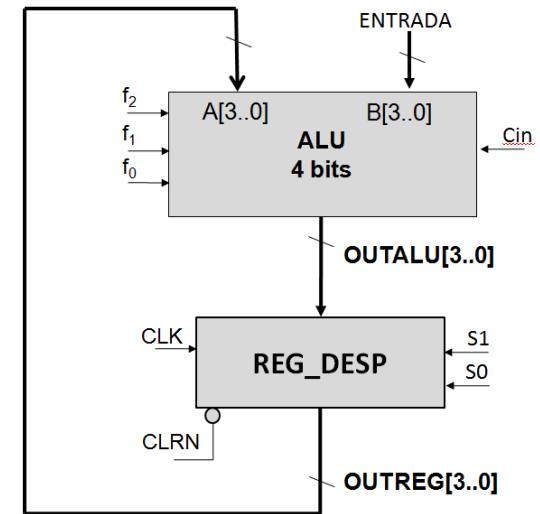
$$W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.

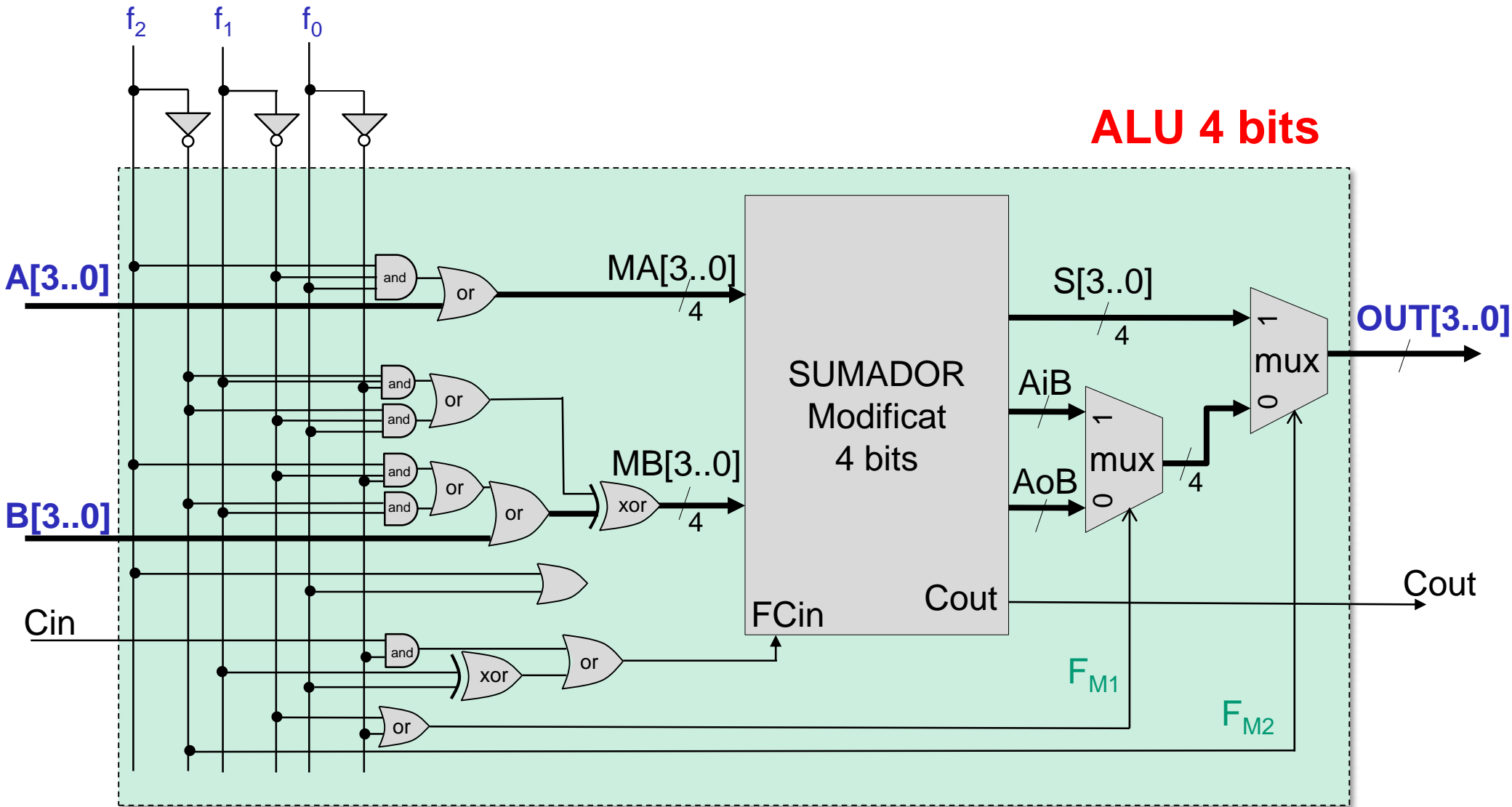
f2	f1	f0	Operació	Descripció operació
0	0	0	A + B + Cin	Suma aritmètica (operació aritmètica)
0	0	1	A - B	Resta aritmètica (operació aritmètica)
0	1	0	A + 1	Increment d'A en una unitat (operació aritmètica)
0	1	1	A - 1	Decreixement de A en una unitat (operació aritmètica)
1	0	0	A	Transferir A (operació lògica)
1	0	1	B	Transferir B (operació lògica)
1	1	0	A and B	AND (operació lògica)
1	1	1	A or B	OR (operació lògica)



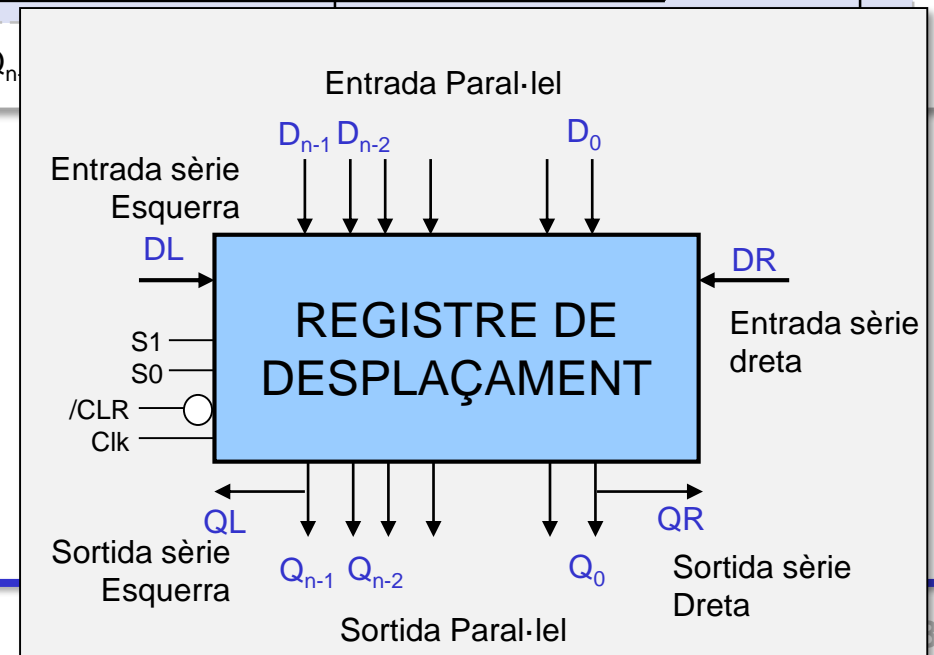
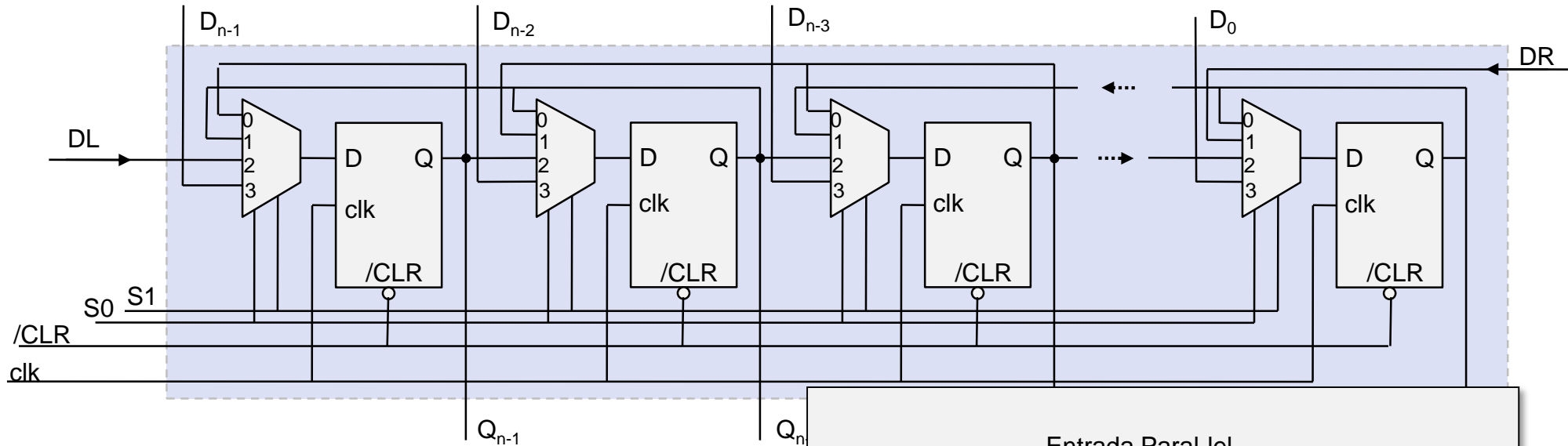
Taula funcionament ALU



Unitat de Procés



Unitat de Procés



Selecció S1 S0	Operació		Senyals de Sortida				
			Q_{n-1}	Q_{n-2}	Q_1	Q_0
0 0	Mantenir	HOLD	Q_{n-1}	Q_{n-2}	Q_1	Q_0
0 1	Despl. esquerra	SHL	Q_{n-2}	Q_{n-3}	Q_0	DR
1 0	Despl. Dreta	SHR	DL	Q_{n-1}	...	Q_2	Q_1
1 1	Càrrega	LOAD	D_{n-1}	D_{n-2}	D_1	D_0

Unitat de Procés

Implementarem una **CPU** senzilla que realitza la següent operació →

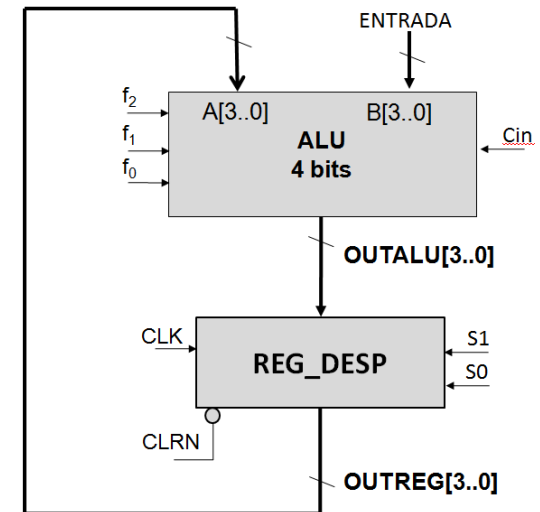
$$W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.

f2	f1	f0	Operació	Descripció operació
0	0	0	A + B + Cin	Suma aritmètica (operació aritmètica)
0	0	1	A - B	Resta aritmètica (operació aritmètica)
0	1	0	A + 1	Increment d'A en una unitat (operació aritmètica)
0	1	1	A - 1	Decreixement de A en una unitat (operació aritmètica)
1	0	0	A	Transferir A (operació lògica)
1	0	1	B	Transferir B (operació lògica)
1	1	0	A and B	AND (operació lògica)
1	1	1	A or B	OR (operació lògica)



Taula funcionament ALU



Senyals de control		Funcionalitat	Senyals de sortida dels FFs			
S1	S0		Q3	Q2	Q1	Q0
0	0	Manté dada	Q3	Q2	Q1	Q0
0	1	Desplaça a l'esquerra	Q2	Q1	Q0	DR
1	0	Desplaça a la dreta	DL	Q3	Q2	Q1
1	1	Càrrega en paral·lel	D3	D2	D1	D0

Taula funcionament Registre de desplaçament →



Unitat de Procés

Implementarem una **CPU** senzilla que realitza la següent operació →

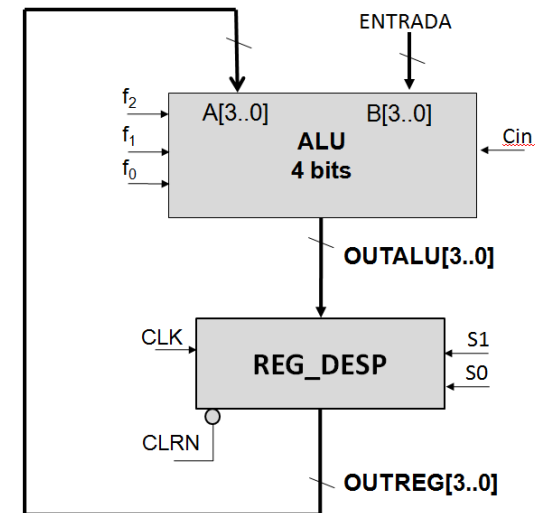
$$W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.

f2	f1	f0	Operació	Descripció operació
0	0	0	A + B + Cin	Suma aritmètica (operació aritmètica)
0	0	1	A - B	Resta aritmètica (operació aritmètica)
0	1	0	A + 1	Increment d'A en una unitat (operació aritmètica)
0	1	1	A - 1	Decreixement de A en una unitat (operació aritmètica)
1	0	0	A	Transferir A (operació lògica)
1	0	1	B	Transferir B (operació lògica)
1	1	0	A and B	AND (operació lògica)
1	1	1	A or B	OR (operació lògica)



Taula funcionament ALU



Senyals de control		Funcionalitat	Senyals de sortida dels FFs			
S1	S0		Q3	Q2	Q1	Q0
0	0	Manté dada	Q3	Q2	Q1	Q0
0	1	Desplaça a l'esquerra	Q2	Q1	Q0	0
1	0	Desplaça a la dreta	0	Q3	Q2	Q1
1	1	Càrrega en paral·lel	D3	D2	D1	D0

Taula funcionament Registre de desplaçament →



Unitat de Procés

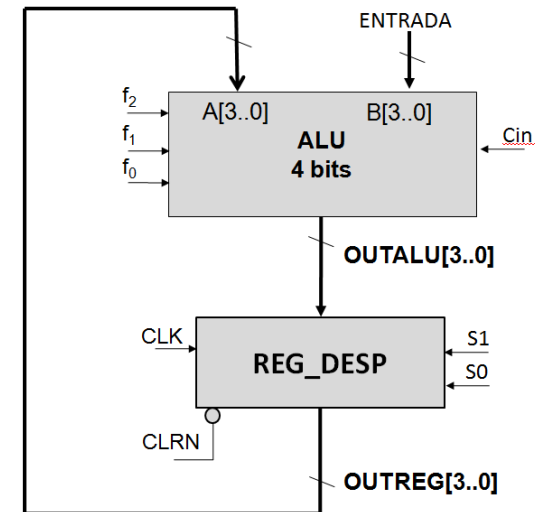
Implementarem una **CPU** senzilla que realitza la següent operació

$$\rightarrow W = \frac{2M + N - P + 1}{4}$$

Amb l'ALU es faran les sumes i restes i la multiplicació per 2 i la divisió per 4 es poden fer realitzant desplaçaments a l'esquerra i a la dreta amb el registre. A aquest circuit que realitza aquesta operació se l'anomena **Unitat de Procés (UP)**.

Per fer funcionar la **UP** caldrà controlar els diversos controls seguint una seqüència determinada tal i com es mostra a la taula:

Operació	DL	DR	f ₂	f ₁	f ₀	S1	S0	Cin	ENTRADA	Sortida al Registre
Carregar M	0	0	1	0	1	1	1	x	M	M
Mult. M per 2	0	0	x	x	x	0	1	x	xxxx	2M
Sumar N	0	0	0	0	0	1	1	0	N	2M + N
Restar P	0	0	0	0	1	1	1	x	P	2M + N - P
Sumar 1	0	0	0	1	0	1	1	x	xxxx	2M + N - P + 1
Dividir per 2	0	0	x	x	x	1	0	x	xxxx	(2M + N - P + 1)/2
Dividir per 2	0	0	x	x	x	1	0	x	xxxx	2M + N - P + 1)/2/2



Aquesta tasca és realitzarà amb una **Unitat de Control (UC)** que generarà seqüencialment tots els senyals de control necessaris tal i com surten a la taula.

Unitat de Control

Implementarem una **CPU** senzilla que realitza la següent operació



$$W = \frac{2M + N - P + 1}{4}$$

Ara l'objectiu és fer una **UC microprogramada** que generi automàticament els senyals de control de l'ALU i del registre i la senyal Cin externa (amb la taula es pot veure que en aquest cas es pot fixar a 0), de manera que només s'han d'introduir els operands i generar el senyal de rellotge. Cada pas a realitzar per executar la operació es pot considerar una micro-operació. L'estructura d'una **microinstrucció (UINST)** serà la següent:

UINST[7] = DL
UINST[6] = DR
UINST[5] = f2
UINST[4] = f1
UINST[3] = f0
UINST[2] = S1
UINST[1] = S0
UINST[0] = Cin

Operació	DL	DR	f2	f1	f0	S1	S0	Cin	ENTRADA	Sortida al Registre
Carregar M	0	0	1	0	1	1	1	x	M	M
Mult. M per 2	0	0	x	x	x	0	1	x	xxxx	2M
Sumar N	0	0	0	0	0	1	1	0	N	2M + N
Restar P	0	0	0	0	1	1	1	x	P	2M + N - P
Sumar 1	0	0	0	1	0	1	1	x	xxxx	2M + N - P + 1
Dividir per 2	0	0	x	x	x	1	0	x	xxxx	(2M + N - P + 1)/2
Dividir per 2	0	0	x	x	x	1	0	x	xxxx	2M + N - P + 1)/2/2

Unitat de Control

Implementarem una **CPU** senzilla que realitza la següent operació



$$W = \frac{2M + N - P + 1}{4}$$

El valor de cada microinstrucció per cada pas a realitzar dependrà per tant dels senyals de control. Aquests s'han d'introduir en una memòria de 16x8 bits de manera que cada posició de memòria contingui una microinstrucció. Com que cada posició de memòria és de 8 bits (**1 byte**) els utilitzarem tots per codificar la microinstrucció. Per exemple, la primera microinstrucció serà "00101110".

UNIST[7] = DL
UNIST[6] = DR
UNIST[5] = f2
UNIST[4] = f1
UNIST[3] = f0
UNIST[2] = S1
UNIST[1] = S0
UNIST[0] = Cin

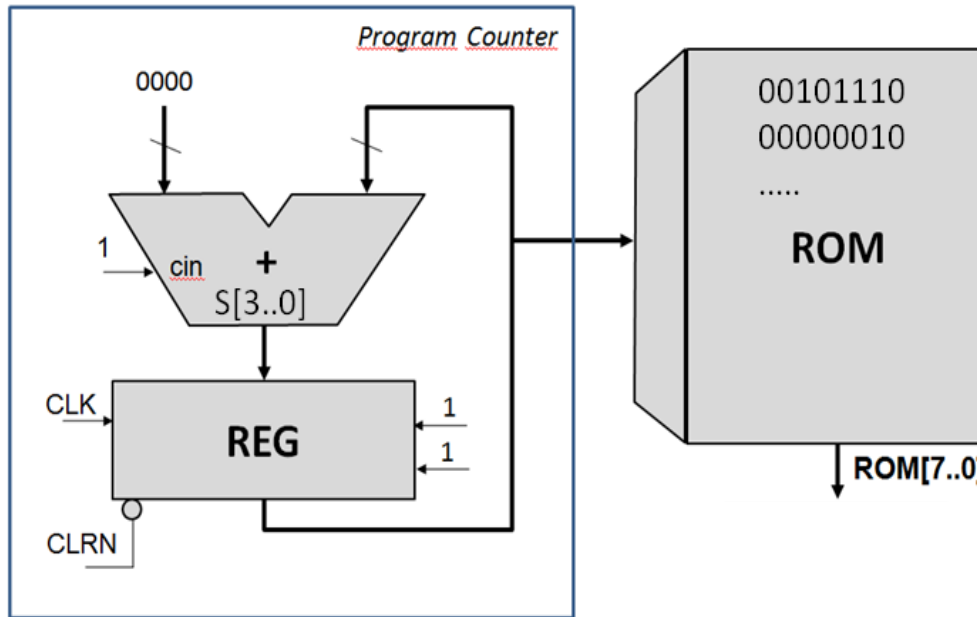
Operació	DL	DR	f2	f1	f0	S1	S0	Cin	ENTRADA	Sortida al Registre
Carregar M	0	0	1	0	1	1	1	0	M	M
Mult. M per 2	0	0	0	0	0	0	1	0	xxxx	2M
Sumar N	0	0	0	0	0	1	1	0	N	2M + N
Restar P	0	0	0	0	1	1	1	0	P	2M + N - P
Sumar 1	0	0	0	1	0	1	1	0	xxxx	2M + N - P + 1
Dividir per 2	0	0	0	0	0	1	0	0	xxxx	(2M + N - P + 1)/2
Dividir per 2	0	0	0	0	0	1	0	0	xxxx	2M + N - P + 1)/2/2

Unitat de Control

Implementarem una **CPU** senzilla que realitza la següent operació

$$\rightarrow W = \frac{2M + N - P + 1}{4}$$

Per seqüencialitzar les microinstruccions una rere l'altra s'utilitzarà un circuit generador d'adreces de la memòria ROM que s'anomena *Program Counter* (PrC). Aquest circuit funciona com a **comptador**, i conté un registre on s'emmagatzema l'adreça de la memòria ROM. A cada cicle de rellotge, el PrC s'incrementa en una unitat de manera que les microinstruccions surten de forma seqüencial i per tant a cada cicle s'executarà una nova operació en la unitat de procés. Per fer el PrC es pot utilitzar un registre i la sortida S (suma) d'un sumador complet modificat de 4 bits.



```
module MEMROM(address, data);
input [3:0] address;
output [7:0] data;
wire [7:0] mem[0:15];
assign data = mem[address]; //contingut de la memoria
assign mem[0] = 8'b00101110;
assign mem[1] = 8'b00000010;
assign mem[2] = 8'b00000110;
assign mem[3] = 8'b00001110;
assign mem[4] = 8'b00010110;
assign mem[5] = 8'b00000100;
assign mem[6] = 8'b00000100;
assign mem[7] = 8'b0;
assign mem[8] = 8'b0;
assign mem[9] = 8'b0;
assign mem[10] = 8'b0;
assign mem[11] = 8'b0;
assign mem[12] = 8'b0;
assign mem[13] = 8'b0;
assign mem[14] = 8'b0;
assign mem[15] = 8'b0;
endmodule
```

Implementarem una **CPU** senzilla que realitza la següent operació



$$W = \frac{2M + N - P + 1}{4}$$

