

Aritmètica d'enters

Tema 3

Curs 2016-2017. Grup 2A

1 Suma d'enters

La Figura 1 mostra un sumador del tipus *Carry Select* per a nombres de 32 bits.

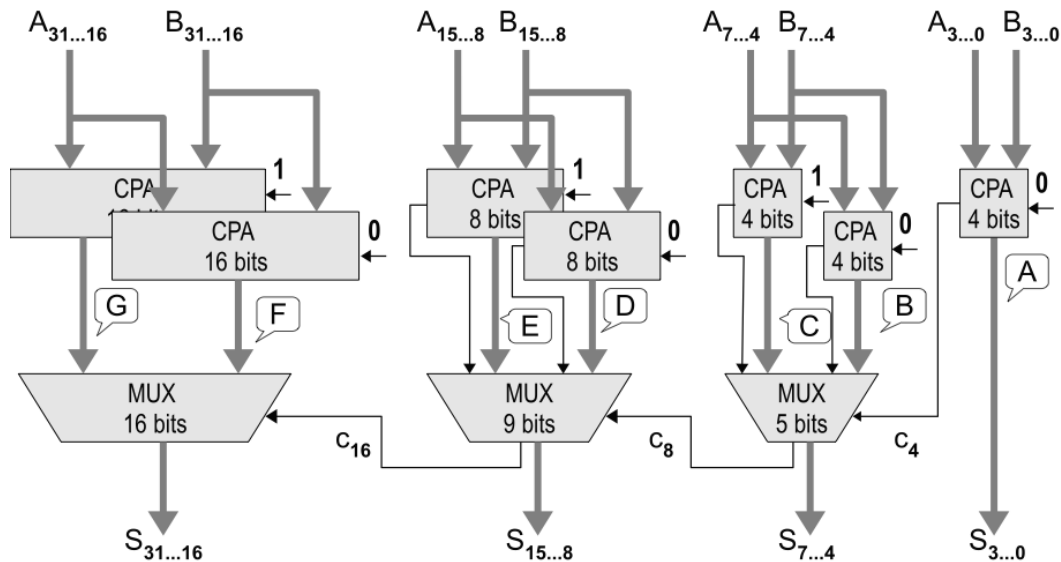


Figura 1: Sumador *Carry Select* per a 32 bits

- Suposeu que els sumadors CPA estan formats per sumadors complets FA amb retards de 1,5 ns per a l'eixida de suma i 1 ns per al transport. El retard dels multiplexors és de 3 ns.

Calculeu en ns el retard en cadascun dels punts que s'indiquen en la taula següent, a partir de l'instant en què s'apliquen les entrades $A_{31...0}$ i $B_{31...0}$. En els punts d'observació B...G indiqueu el retard del bit més significatiu (msb).

S_0	S_3	c_4	B(msb)	C(msb)	S_7	c_8	D(msb)	E(msb)	S_{15}	c_{16}	F(msb)	G(msb)	S_{31}

Quina és la productivitat de l'operador? Expresseu-la en MOPS.

- Preneu ara el vostre DNI (sense la lletra final, és clar) i el vostre IND.

El vostre IND és el DNI amb les xifres en ordre invers. Per exemple: si vostre DNI fóra 456789123, el vostre IND seria 321987654.

Escriviu ambdós valors en decimal i en hexadecimal:

DNI (dec)	DNI (hex)	IND (dec)	IND (hex)

Apliqueu a l'operador els operands $A = \text{DNI}$ i $B = \text{IND}$ i calculeu (en hexadecimal) els valors indicats en aquesta taula:

$S_{3...0}$	c_4	B	C	$S_{7...4}$	c_8	D	E	$S_{15...8}$	c_{16}	F	G	$S_{31...16}$

2 Multiplicació en assembler per una constant

Heu d'escriure el codi de les funcions `word multK1(word a)` i `word multK2(word a)` en assembler del MIPS R2000 que multiplique el seu únic argument per les constants K_1 i K_2 .

Obtenció de K_1 i K_2

- La primera constant és positiva, i la calculeu com $K_1 = \text{DNI} \% 1000$, és a dir, les tres xifres menys significatives del vostre DNI.
Si la constant així calculada resulta $K_1 < 100$, féu $K_1 = K + 500$.
- La segona constant serà negativa. $K_2 = -K_1$

Exemples:

- Si el vostre DNI és 234567890-M, $K_1 = 890$ i $K_2 = -890$
- Si el vostre DNI és 234567098-M, $K_1 = 598$ i $K_2 = -598$

1. Escriviu les constants K_1 i K_2 en decimal, en complement a 2 (12 bits) i amb la codificació de Booth.

Ompliu una taula com aquesta:

	Decimal	Binari	Booth
K_1			
K_2			

2. Escriviu el codi de la funció `word multK1(word a)` en assembler del MIPS R2000. No podeu fer ús de les instruccions de multiplicació del MIPS: heu de treballar amb només sumes i desplaçaments, basant-vos en la representació binària del multiplicador.
3. Escriviu el codi de la funció `word multK2(word a)` en assembler del MIPS R2000. No podeu fer ús de les instruccions de multiplicació del MIPS, heu de treballar amb sumes, restes i desplaçaments, basant-vos en la recodificació del multiplicador per Booth

En els apartats 2 i 3 seguiu el conveni d'escriptura de les funcions en assembler: la funció prendrà l'argument del registre `$a0` i desarà el resultat en `$v0`.