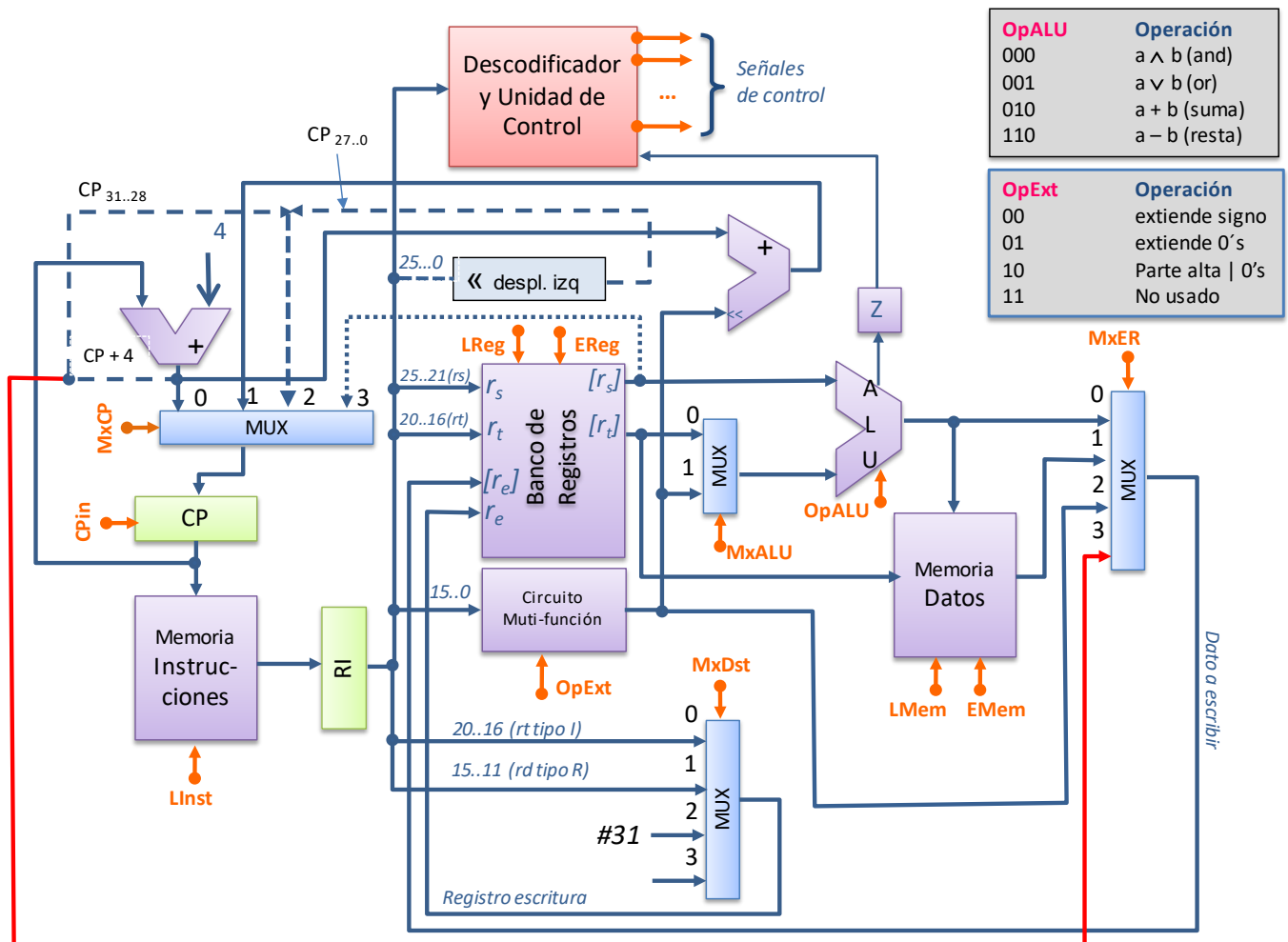


Apellidos y Nombre

DNI

Grupo

- 1 (2.0 puntos)** La figura 1 muestra la ruta de datos monociclo del procesador MIPS R2000 modificada para dar soporte a más instrucciones. Para ello se han ampliado algunos multiplexores, para abrir nuevos caminos, y se han mejorado algunos elementos, como el circuito multifunción para el tratamiento del operando inmediato en instrucciones de tipo I. Así se puede, según la señal **OpExt**, extender el signo del operando a 32 bits, o extender con ceros o poner el valor inmediato en la parte alta y ceros la parte baja. Las distintas opciones de este circuito aparecen también en la Figura 1. Los nuevos caminos añadidos permiten ejecutar la instrucción *jump* (**J etiqueta** - - línea discontinua) y la instrucción *jump register* (**Jr \$rs ...** línea punteada).



Se pretende ahora incorporar nuevas instrucciones, como son la JAL (Jump and Link) y la JALR (Jump and Link Register):

- **JAL etiqueta** Jump and Link (Salto a subrutina)

Guarda en \$31 la dirección de retorno (CP+4) y salta a la dirección apuntada por etiqueta. Esta dirección se obtiene de la misma forma que la instrucción JUMP (4 bits del CP actual || operando inmediato 26 bits || 00).

$$\$31 \leftarrow CP+4$$

$$CP \leftarrow \text{dirección etiqueta}$$

- **JALR rd, rs** Jump and Link Register (Salto a subrutina apuntada por rs)

Guarda en rd la dirección de retorno (CP+4) y salta a la dirección apuntada por rs.

$$rd \leftarrow CP+4$$

$$CP \leftarrow rs$$

a) (0.4 puntos) Complete la figura marcando las conexiones necesarias para poder ejecutar las dos nuevas instrucciones.

b) (0.4 puntos) Complete la tabla correspondiente a las señales de control para ejecutar las nuevas instrucciones:

Instrucción	Form	Ereg	OpALU	LMem	EMem	OpExt	MxPC	MxALU	MxER	MxDst
jal etiqueta	J	0	XXX	0	0	XX	10	X	11	10
jalr rd, rs	J	1	XXX	0	0	XX	11	X	11	01

c) (0.7 puntos) Complete la tabla correspondiente a las señales de control para ejecutar en la misma ruta de datos las instrucciones que se indican:

Instrucción	Form	Ereg	OpALU	LMem	EMem	OpExt	MxPC	MxALU	MxER	MxDst
add rd,rs,rt	R	1	010	0	0	XX	00	0	00	01
ori rt, rs, inmed	I	1	001	0	0	01	00	1	00	00
lw rt, desp(rs)	I	1	010	1	0	00	00	1	01	00
bne rs, rt, etiq.	I	0	110	0	0	00	0Z*	0	XX	XX
lui rt, inmed	I	1	XXX /010	0	0	10	00	X / 1	10 / 00	00
j etiqueta	J	0	XXX	0	0	XX	10	X	XX	XX

Supóngase los siguientes retardos en los elementos que componen esta ruta de datos: lectura/escritura en registros 5 ns, operación en la ALU 10 ns, circuito multifunción 2 ns, el resto de retardos se considera despreciable, excepto las memorias.

d) (0.5 puntos) Si se pretende que la productividad máxima de este procesador sea de 25 MIPS, ¿cuál debería ser la velocidad de las memorias (retardo en ns)? Justifique la respuesta.

Productividad = 25 MIPS

La productividad es una instrucción por ciclo. Luego la frecuencia del reloj es de 25 MHz (40 ns)

El periodo de reloj T tiene que permitir ejecutar la instrucción más larga (load word).

$T = T_{mem} + 5 + 10 + T_{mem} + 5 = 40 \text{ ns} \rightarrow T_{mem} = 10 \text{ ns}$

**2** (1 punto) Suponga un procesador monociclo como el del ejercicio anterior, pero con los siguientes componentes: Memorias (20ns lect/escrit), ALU (14ns), banco de registros (8ns lect/escrit), circuito multifunción (4ns). Para aumentar la productividad de este procesador, se segmenta en 5 etapas (LI, DI, EX, M, ER). Asumiendo que el retardo de los registros de segmentación es de 5ns, se pide:

a) (0,2 puntos) Frecuencia de reloj del procesador segmentado (Justifique la respuesta indicando los cálculos).

$\tau = \max(20, 8, 14, 20, 8) + 5 = 25 \text{ ns}$

$F = 1 / 25 \text{ ns} = 40 \text{ MHz}$

- b) (0,4 puntos) Productividad máxima del procesador segmentado y aceleración máxima respecto del procesador original monociclo (Justifique la respuesta indicando los cálculos).

El periodo de reloj de procesador monociclo T tiene que permitir ejecutar la instrucción más larga (load word)

$$T = T_{\text{mem}} + T_{\text{reg}} + T_{\text{alu}} + T_{\text{mem}} + T_{\text{reg}} = 20 + 8 + 14 + 8 + 20 = 70\text{ns}$$

El periodo del segmentado es:  $\tau = 25\text{ns}$

Productividad máxima  $\chi(\infty) = 1/\tau = 40\text{ MIPS}$

Aceleración máxima  $S(\infty) = T/\tau = 70/25 = 2.8$

Aceleración Ideal  $S_{\text{ideal}} = k = 5$

- c) (0,4 puntos) Para mejorar el diseño anterior se procede a supersegmentar el procesador, dividiendo las etapas LI y M en dos subetapas de 10ns (LI1, LI2, DI, EX, M1, M2, ER) y empleando registros de etapa de 2ns. El resto de etapas son igual al anterior. Además, se replica el conjunto 4 veces haciendo un procesador super-escalar de grado 4. (IPC = 4). Indique la productividad máxima de este nuevo procesador super-escalar y la aceleración máxima respecto del procesador original monociclo no segmentado (Justifique la respuesta).

El periodo de reloj de procesador monociclo  $T = 70\text{ns}$

Periodo de reloj del procesador super-escalar:

$$\tau = \max(10, 10, 8, 14, 10, 8) + 2 = 16\text{ns}$$

Productividad máxima  $\chi(\infty) = 4\text{ instrucciones/ciclo de reloj} = 4/16 = 250\text{ MIPS}$

Aceleración máxima  $S(\infty) = \chi_{\text{super-seg}} / \chi_{\text{no-seg}} = (4/\tau) / (1/T) = 4T/\tau = 4 \times 70/16 = 17.5$

### 3 (1.0 punto) Se va a ejecutar la siguiente subrutina en ensamblador del MIPS R2000.

```
(1) Subrut: addi $t2, $0, 50
(2)          ori $t1, $t1, 0
(3) bucle: lw $t0, 0($a0)          # bucle 50 iteraciones
(4)          add $t1, $t1, $t0
(5)          addi $t2, $t2, -1
(6)          addi $a0, $a0, 4
(7)          bnez $t2, bucle
(8)          sw $t1, 0($a0)
(9)          jr $ra          # retorno de subrutina
```

Asuma que los conflictos por dependencias de datos se solucionan mediante la inserción de ciclos de parada. Así mismo, para los riesgos de control en instrucciones de salto condicional se emplea la técnica de Predicción de salto no efectivo (Predict Not Taken). Para los riesgos en instrucciones de salto incondicional se emplean ciclos de parada. La latencia de salto es 2. Se asume que el registro \$a0 ha sido previamente inicializado.

- a) (0.3 puntos) Identifique los conflictos por dependencia de datos que se producen en dicho código. Rellene para ello la tabla siguiente utilizando tantas filas como necesite:

	Registro	Número de instrucción en que se escribe	Número de instrucción en que se lee
Riesgo 1	\$t1	2	4
Riesgo 2	\$t0	3	4
Riesgo 3	\$t2	5	7
Riesgo 4	\$a0	6	8
Riesgo 5			

- b) (0,3 puntos) Complete el diagrama instrucciones/ciclo para las instrucciones indicadas que se ejecutan en la última vuelta de bucle.

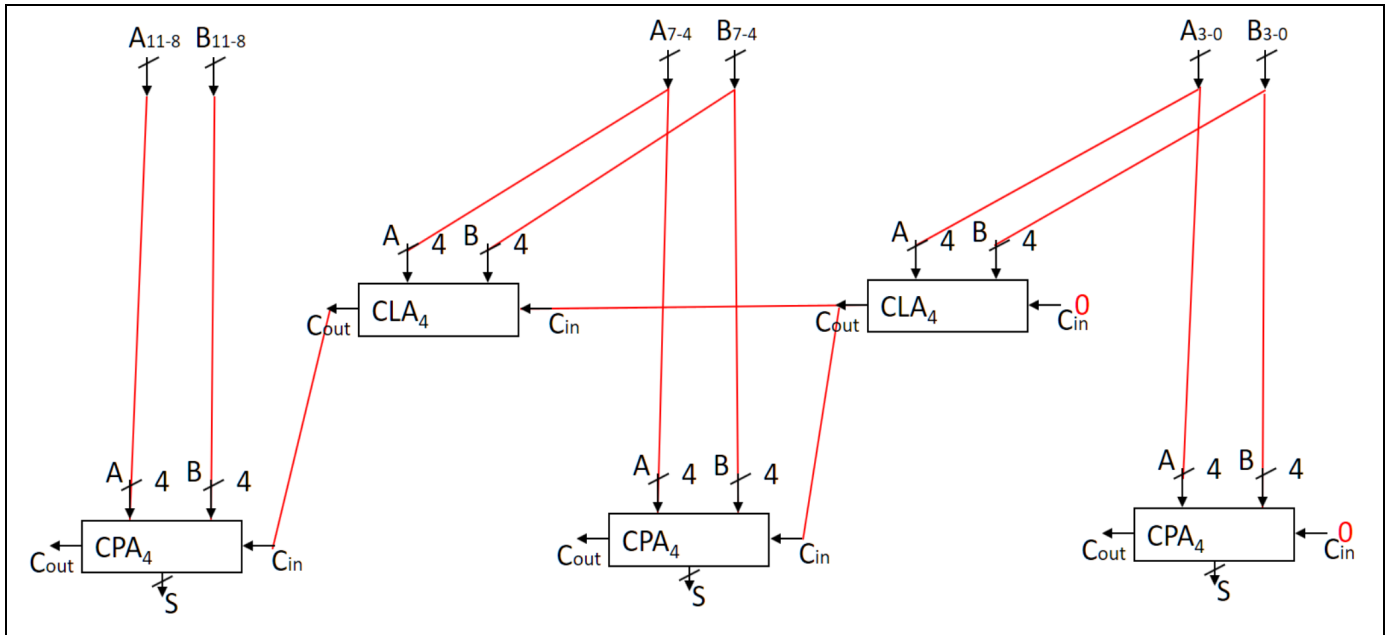
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw \$t0, 0(\$a0)	LI	DI	EX	M	ER	\$t0									
add \$t1, \$t1, \$t0		LI	DI	DI	DI	EX	M	ER							
addi \$t2, \$t2, -1			LI	LI	LI	DI	EX	M	ER	\$t2					
addi \$a0, \$a0, 4						LI	DI	EX	M	ER	\$a0				
bnez \$t2, bucle							LI	DI	DI	EX	M	ER			
sw \$t0, 0(\$a0)								LI	LI	DI	EX	M	ER		

- c) (0,4 puntos) Calcule los datos siguientes para la ejecución completa del código de la subrutina justificando siempre los valores

Número de Instrucciones ejecutadas (I)	$2 + 5 \cdot 50 + 2 = 254$
Número de ciclos de parada (P)	Hay 2 ciclos por la dependencia (3)-(4), 1 por la (5)-(7) que se repiten 50 veces, 2 por el salto pero sólo en las primeras 49 iteraciones del bucle donde la predicción falla. En la última iteración no falla. Finalmente 2 por la instrucción de retorno (jr \$ra)
Número de ciclos totales de ejecución (C)	$C = I + P + 4 = 254 + 250 + 4 = 508$
CPI	$CPI = (C-4) / I = 1 + P/I = 1 + 250/254 = 1,9843$

**4 (1 punto)** Se desea desarrollar un sumador de 12 bits utilizando tres sumadores con propagación de acarreo de 4 bits (con un tiempo de 9ns para la suma del último bit y de 8 ns para el acarreo de salida) y con dos circuitos derivados de los CLA (*Carry Lookahead Adder*) de 4 bits que son capaces de calcular el acarreo de la suma de dos números de 4 bits mas un acarreo de entrada en solo 3ns.

a) (0,7 puntos) Dibuje las líneas de conexión de los datos de entrada con los circuitos y de los circuitos entre sí para llevar a cabo dicha operación.



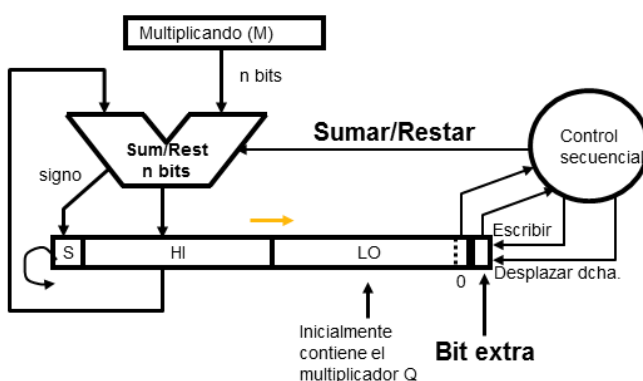
b) (0,3 puntos) Calcule el tiempo que tardará en completarse la suma de 12 bits y compárelo con el que tardaría el mismo circuito sin hacer uso de los CLA. Indique la mejora (speedup).

El tiempo del circuito viene dado por la salida de la línea S11, cada CLA retrasa en 3 ns el cálculo de la entrada Cin del último CPA, por tanto 6 ns que sumaremos a los 9ns, por lo que tardará 15 ns.

La opción con solo CPAs necesitaría 8ns para calcular el acarreo de salida de cada CPA, por tanto, aquí tardaría  $8ns + 8ns + 9ns = 25ns$

El speedup será ( $25\text{ns} / 15\text{ns} = 1,67$ )

**5 (1 punto)** Se dispone de un circuito multiplicador secuencial basado en el **algoritmo de Booth** para números enteros de 4 bits tal y como se muestra en la figura. Supóngase que el retardo de las diferentes operaciones involucradas en la multiplicación es el que se indica en la tabla adjunta:



Inicializar registros y circuito de control	5 ns
Inspeccionar $q_i, q_{i-1}$	2 ns
Sumar y actualizar HI	8 ns
Restar y actualizar HI	11 ns
Desplazar S-HI-LO-X 1 bit	2 ns
Evaluar el número de ciclo actual	1 ns

a) (0.7 puntos) Indique paso a paso como se realizaría el producto de los números de 4 bits M=2 y Q=-7.

Ciclo	Acción	S-HI-LO-X
0	Valores iniciales	0 0000 <b>1001</b> <u>0</u>
1	Caso 10: $HI \leftarrow HI - M$	1 1110 <b>1001</b> 0
	Desplazar S-HI-LO 1 bit a la derecha	1 1111 0 <b>100</b> <u>1</u>
2	Caso 01: $HI \leftarrow HI + M$	0 0001 0 <b>100</b> 1
	Desplazar S-HI-LO 1 bit a la derecha	0 0000 10 <b>10</b> <u>0</u>
3	Caso 00: No hacer nada	0 0000 10 <b>10</b> 0
	Desplazar S-HI-LO 1 bit a la derecha	0 0000 010 <b>1</b> <u>0</u>
4	Caso 10: $HI \leftarrow HI - M$	1 1110 010 <b>1</b> 0
	Desplazar S-HI-LO 1 bit a la derecha	<b>1</b> 1111 0010 1

b) (0.3 puntos) ¿Cuál sería la productividad que podría alcanzar dicho multiplicador, en MOPS? Considere que la inicialización se realiza en un ciclo de reloj independiente. Justifique la respuesta.

Tiempo de operación =  $\max(5ns, (2ns+11ns+2ns+1ns)) \cdot (1+4) = 80ns$

Productividad =  $1/80ns = 12,5 \text{ MOPS}$

**6** (1 punto) Dada la siguiente secuencia de instrucciones en un lenguaje de alto nivel. complete el recuadro de la derecha con el código del ensamblador MIPS que permite realizar dicha tarea.

<pre>float PI = 3.1416; double r3 = 2.344*2.344*2.344; double K = 4.0 / 3.0; double V; // Cálculo V = K* (double)PI* r3;</pre>	<pre>.data 0x10000000 PI : .float 3.1416 r3:  .double 2.344*2.344*2.344 K:  .double 4.0/3.0 V:  .space 8 .text 0x00400000 __start: la \$t0, PI         lwc1 \$f0, 0(\$t0)      # o también l.s \$f0, PI         cvt.d.s \$f0, \$f0      # (double)PI         la \$t0, r3         lwc1 \$f2, 0(\$t0)         lwc1 \$f3, 4(\$t0)      # \$f3   \$f2 &lt;- r3         la \$t0, K         lwc1 \$f4, 0(\$t0)         lwc1 \$f5, 4(\$t0)      # \$f5   \$f4 &lt;- K         mul.d \$f0, \$f0, \$f2    # PI*r3         mul.d \$f0, \$f0, \$f4    # K*PI*r3         la \$t0, V         swc1 \$f0, 0(\$t0)         swc1 \$f1, 4(\$t0)      # V &lt;- \$f1   \$f0         .end</pre>
--	---

**7 (1.5 puntos)** Un módulo M1 de 1GB utilizado en un procesador de 32 bits está implementado mediante chips comerciales de memoria SDRAM cuyo ancho de palabra es de 8 bits.

a) (0.5 puntos) Se observa externamente que el módulo tiene dos filas de chips. ¿De qué capacidad será cada uno de ellos? Justifique la respuesta.

Al tener chips de 8 bits necesitamos 4 (32/8) por fila y por tanto habrá 8 chips de capacidad  $1\text{GB}/8 = 128\text{MB}$

b) (0.25 puntos) Los chips indicados tienen 4 bancos, y se sabe que hay 1024 filas. ¿Cuántas columnas tendrá cada banco? Justifique la respuesta.

$\text{Columnas} = 128\text{M} / (4 * 1024) = 2^{27} / 2^{12} = 2^{15} = 32\text{K}$  filas

c) (0.25 puntos) ¿Cuántas patillas para las direcciones ( $A_i$ ) tendrá cada chip? Justifique la respuesta

$\max(10, 15) = 15$  líneas

d) (0.25 puntos) Y el módulo M1 ¿cuántas líneas  $DMQ_i$  tendrá? Justifique la respuesta.

4 una por byte

e) (0,25 puntos) Si la frecuencia del bus del sistema es 1GHz. ¿Cuál sería el ancho de banda de cada chip? ¿y el del módulo M1? Justifique la respuesta

Chip (GBps) =  $f * 1 \text{ Byte} = 1\text{GBps}$

Módulo M1 (GBps) =  $f * 4 \text{ Bytes} = 4\text{GBps}$

**8 (1,5 puntos)** A un computador de 32 bits, con 28 bits para las direcciones, se le ha dotado de un primer módulo M1 de 16MB a partir de la dirección 0 y otro módulo M2 de igual tamaño justo a continuación.

a) (0.5 puntos) Complete las direcciones inicial y final de cada módulo

Módulo M1		Módulo M2	
Dirección Inicial	Dirección final	Dirección Inicial	Dirección final
0x0000000	0x0FFFFFF	0x1000000	0x1FFFFFF
Función de selección a nivel bajo:		Función de selección a nivel bajo:	
A27+A26+A25+A24		A27+A26+A25+A24*	

a) Los módulos utilizan chips DDR3-SDRAM a 1GHz con tamaño de bloque de memoria de 8 palabras y su temporización es  $C_L-t_{rcd}-t_{ras}-t_{rp}=3-2-5-3$ .

b1) (0.5 puntos) Calcule:

Periodo de reloj = $1/1\text{GHz} = 1\text{ns}$
tacc= trcd + tCL= 5 ciclos = 5ns
Ancho de banda (BW)= al ser DDR es el doble de la frecuencia = $2 \times 1\text{GHz} \times 4\text{bytes} = 8\text{GBps}$

b2) (0.5 puntos) Rellene sobre el casillero siguiente las órdenes a realizar para leer dos bloques enteros consecutivos, almacenados en la misma fila. La precarga es automática.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21
Orden	A		L1				L2														
Dirección	F		C				C														
Datos						D0D1	D2D3	D4D5	D6D7	D0D1	D2D3	D4D5	D6D7								