

LABORATORIO DE ARQUITECTURA DE COMPUTADORES

REPERTORIO ESENCIAL DEL PROCESADOR DLX

Características generales

El procesador DLX es una arquitectura RISC de propósitos didácticos cuyo nombre se obtiene promediando las denominaciones de diferentes procesadores comerciales y experimentales, expresado en números romanos (DLX=560). Este procedimiento imita el que utilizó Knuth para el procesador MIX, explicado en el clásico libro *The Art of Computer Programming* y los padres del DLX le hacen un guiño a este autor aplicándolo de un modo similar a este procesador.

En este documento es un extracto de otro más completo y solo exponemos las instrucciones esenciales para hacer las prácticas diseñadas.

Las características más importantes del DLX son:

1. Utiliza 32 Registros de Propósito General (GPR) para enteros de 32 bits: R0 a R31. La palabra es, por lo tanto, de 32 bits (4 bytes). **R0 es siempre 0** y no admite cargar ningún valor.
2. Tiene 32 registros de coma flotante (FP por Floating Point) denominados F0 a F31 de 32 bits. Estos pueden funcionar como registros individuales de Simple Precisión (SP) o en parejas de Doble Precisión (64 bits, DP) en cuyo caso se les referencia con los números pares de los registros (F0, F2, F4...F30). El formato de representación de coma flotante es el IEEE754.
3. La memoria se direcciona por bytes, en modo BIG ENDIAN (OJO). Las direcciones son de 32 bits de largo (bus de direcciones), y todas las instrucciones ocupan ese mismo tamaño (una palabra) y están alineadas a palabras.

Instrucciones de transferencia de Datos (LOAD STORE)

Transfieren datos **entre**

- Registros y Memoria, donde la dirección de memoria es el contenido de un registro GPR + desplazamiento inmediato de 16 bits

LW	SW	Carga palabra, Almacena palabra (a/desde registros enteros)
LD	SD	Carga coma flotante DP, Almacena coma flotante DP

Instrucciones de proceso (enteros)

Operaciones aritméticas (excepto multiplicación y división) y lógicas con datos enteros o lógicos en registros GPR/s. En operaciones con signo, en caso de rebose (overflow) se genera un TRAP. Se opera entre registros o entre un registro y un dato inmediato. Todos los datos inmediatos son de 16 bits.

ADD / ADDI	Suma entera entre registros / Suma entera con inmediato;
SUB / SUBI	Resta entera entre registros / Resta entera con inmediato;
AND / OR / XOR	AND OR y XOR entre registros
ANDI / ORI / XORI	AND OR y XOR con inmediato
SLL, SRL,	Desplazamientos lógicos a la izquierda (SLL) o derecha (SRL),
SLLI, SRLI,	Desplazamiento con inmediato izquierda o derecha (SLLI, SRLI).
SRA	Desplazamientos aritméticos solo a la derecha
SRAI	Desplazamientos aritméticos solo a la derecha con inmediato

Instrucciones de multiplicación y división (enteros)

Requieren registros FP. En caso de rebose (overflow) se genera un TRAP.

MULT, DIV	Multiplica y divide, con signo y sin signo; los operandos deben estar en registros de <u>coma flotante</u> ; todas las operaciones tienen valores de 32 bits
------------------	--

Instrucciones de Coma Flotante (FP) para DP

ADDD, SUBD	Suma, Resta en DP
MULTD, DIVD	Multiplica, Divide en DP
CVTD2I, CVTI2D	Conversión: CVT _x 2 _y convierte de tipo <u>x</u> a <u>y</u> . I=entero D=DP
ltD, gtD, leD, eqD, neD	Compara registros y pone bit de comparación en registro de estado FP.

Instrucciones de control

Salto y bifurcaciones. Condicionales e incondicionales. Direccionamiento relativo al PC con desplazamiento inmediato o directo de registro.

BEQZ, BNEZ	Salto según GPR sea igual/no igual a cero; desplazamiento de 16 bits desde PC+4
J, JR	Bifurcaciones: despl de 26 bits desde PC (J) o destino en registro (JR)
TRAP	Transfiere a sistema operativo a una dirección vectorizada
BFPT, BFPF	Test (True/False) de Bit de comparación en el registro de estado FP y salto (desplazamiento de 16 bits desde PC+4)

Ejemplos

Notación:

- La numeración de los bits de los registros sigue el modelo BIG ENDIAN:
- el bit 0 es el de la izquierda, el de mayor peso o el de signo
- el bit 31 es el de la derecha, el de menor peso
- ## implica concatenación
- \leftarrow_{32} significa transferencia de 32 bits
- $(R3_0)^{24}$ replica 24 veces el bit 0 de R3

<i>Instrucción</i>	<i>Descripción</i>	<i>Acción</i>
LW R1,30(R2)	Cargar palabra	$R1 \leftarrow_{32} M[30+R2]$ (transferencia de 32 bits)
LW R1,100(R0)	Cargar palabra	$R1 \leftarrow_{32} M[100+0]$
LD F0,50(R2)	Cargar doble	$F0 \## F1 \leftarrow_{64} M[50+R2]$
SW 500(R4),R3	Almacenar palabra	$M[500+R4] \leftarrow_{32} R3$
SD 40(R3),F0	Almacenar flotante DP	$M[40+R3] \leftarrow_{32} F0$; $M[44+R3] \leftarrow_{32} F1$
J dest	Bifurcación	$PC \leftarrow \text{dest}; ((PC+4)-2^{15}) \leq \text{dest} < ((PC+4)+2^{15})$
JR R31	Bifurcación a registro	$PC \leftarrow R31$ (puede implementar un "return")
BEQZ R4, dest	Salta si igual a cero	If $(R4=0)$ $PC \leftarrow \text{dest};$ $((PC+4)-2^{15}) \leq \text{dest} < ((PC+4)+2^{15})$
BNEZ R4, dest	Salta si no igual a cero	If $(R4 \neq 0)$ $PC \leftarrow \text{dest};$ $((PC+4)-2^{15}) \leq \text{dest} < ((PC+4)+2^{15})$
ADD R1,R2,R3	Suma	$R1 \leftarrow R2 + R3$
ADDI R1,R2,#3	Suma con inmediato	$R1 \leftarrow R2 + 3$
SLLI R1,R2,#5	Desplazamiento lógico a la izquierda	$R1 \leftarrow R2 \ll 5$

Trucos y consejos

Si te fijas, la multiplicación de enteros implica el uso de registros de coma flotante. Muy frecuentemente, podemos multiplicar en una fracción del tiempo empleado por esta instrucción si utilizamos en su lugar la operación de desplazamiento a la izquierda. Multiplicar por 2 no es mas que desplazar el operando un lugar a la izquierda: **sll R7,R7,1**

Al hacer un bucle, es más ventajoso llevar un contador descendente, decrementarlo y comprobar si es o no 0 para decidir si se salta al principio del bucle o no.

Para leer datos de una lista, es usual utilizar un registro índice que, sumado a la dirección de comienzo del primer elemento, nos permite acceder a él dentro de un bucle. En este bucle se incrementa por el tamaño del dato (ej: 4 para enteros) para acceder al siguiente elemento en la siguiente iteración.

Es posible combinar la función de contador del bucle e índice usando un único registro. Su valor debe ser 4 veces el del contador inicial y el decremento de 4 en 4 teniendo en cuenta que se accederá del último al primer elemento de la lista (para el caso de enteros, 8 para double etc.).