

Figura 8.15

Conjunto completo de instrucciones de la máquina TINY

### Instrucciones RO

Formato:	<i>opcode r, s, t</i>
<i>Opcode</i>	Efecto
HALT	detener ejecución (operandos ignorados)
IN	$\text{reg}[r] \leftarrow$ lectura de valor entero desde la entrada estándar ( <i>s</i> y <i>t</i> ignorados)
OUT	$\text{reg}[r] \rightarrow$ la salida estándar ( <i>s</i> y <i>t</i> ignorados)
ADD	$\text{reg}[r] = \text{reg}[s] + \text{reg}[t]$
SUB	$\text{reg}[r] = \text{reg}[s] - \text{reg}[t]$
MUL	$\text{reg}[r] = \text{reg}[s] * \text{reg}[t]$
DIV	$\text{reg}[r] = \text{reg}[s] / \text{reg}[t]$ (puede generar ZERO_DIV)

### Instrucciones RM

Formato: *opcode r, d(s)*

( $a = d + \text{reg}[s]$ ; cualquier referencia a  $\text{dMem}[a]$  genera DMEM\_ERR si  $a < 0$  o  $a \geq \text{DADDR\_SIZE}$ )

<i>Opcode</i>	Efecto
LD	$\text{reg}[r] = \text{dMem}[a]$ (carga <i>r</i> con valor de la memoria en <i>a</i> )
LDA	$\text{reg}[r] = a$ (carga dirección <i>a</i> directamente en <i>r</i> )
LDC	$\text{reg}[r] = d$ (carga constante <i>d</i> directamente en <i>r</i> — <i>s</i> es ignorada)
ST	$\text{dMem}[a] = \text{reg}[r]$ (almacena valor en <i>r</i> a localidad de memoria <i>a</i> )
JLT	if ( $\text{reg}[r] < 0$ ) $\text{reg}[\text{PC\_REG}] = a$ (salta a instrucción <i>a</i> si <i>r</i> es negativa, de manera similar para el siguiente)
JLE	if ( $\text{reg}[r] \leq 0$ ) $\text{reg}[\text{PC\_REG}] = a$
JGE	if ( $\text{reg}[r] \geq 0$ ) $\text{reg}[\text{PC\_REG}] = a$
JGT	if ( $\text{reg}[r] > 0$ ) $\text{reg}[\text{PC\_REG}] = a$
JEQ	if ( $\text{reg}[r] == 0$ ) $\text{reg}[\text{PC\_REG}] = a$
JNE	if ( $\text{reg}[r] \neq 0$ ) $\text{reg}[\text{PC\_REG}] = a$

donde los operandos *r*, *s*, *t* son registros legales (verificados en tiempo de carga). De este modo, tales instrucciones son tres direcciones, y las tres direcciones deben ser registros. Todas las instrucciones aritméticas se encuentran limitadas a este formato, como lo están las dos instrucciones primitivas de entrada/salida.

Una instrucción de memoria de registro tiene el formato

*opcode r, d(s)*

En este código *r* y *s* deben ser registros legales (verificados en tiempo de carga), y *d* es un entero positivo o negativo que representa un desplazamiento. Esta instrucción es una instrucción de dos direcciones, donde la primera dirección es siempre un registro y la segunda dirección es una dirección de memoria *a* dada por  $a = d + \text{reg}[r]$ , donde *a* debe ser una dirección legal ( $0 \leq a < \text{DADDR\_SIZE}$ ). Si *a* está fuera del intervalo legal, entonces se genera DMEM\_ERR durante la ejecución.