Reference Card: Jam2000 Instructions

Arithmetic

Encoding	*Fmt*	*Assembly*	*Meaning*
? ? ? z y x 1 0	R	(add x y z)	$Rx \leftarrow Ry + Rz$
? ? ? z y x 2 0	R	(sub x y z)	$Rx \leftarrow Ry - Rz$
? ? ? z y x 3 0	R	(mul x y z)	$Rx \leftarrow Ry \times Rz$
? ? ? z y x 4 0	R	(div x y z)	$Rx \leftarrow Ry \div Rz$
? ? ? z y x 5 0	R	(mod x y z)	$Rx \leftarrow Ry \bmod Rz$

Memory Loads and Stores

Encoding	*Fmt*	*Assembly*	*Meaning*
$\pm d_7 \ldots d_2$ x 9	I	(ldi x $\pm d_7 \ldots d_2$)	$Rx \leftarrow \pm d_7 \ldots d_2$
? ? 1 ? y x 6 0	R	(ld x y)	$Rx \leftarrow mem[Ry]$
? ? 0 z y x 6 0	R	(ldx x y z)	$Rx \leftarrow mem[Ry + Rz]$
? ? 1 ? y x 7 0	R	(st x y)	$mem[Ry] \leftarrow Rx$
? ? 0 z y x 7 0	R	(stx x y z)	$mem[Ry + Rz] \leftarrow Rx$
? ? ? ? y x 9 0	R	(mov x y)	$Ry \leftarrow Rx$

Control

Encoding	*Fmt*	*Assembly*	*Meaning*
? ? ? ? ? x 8 0	R	(jmpx x)	$PC \leftarrow Rx$
$\pm d_7 \ldots d_2$? 1	I	(jmpi $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$
$\pm d_7 \ldots d_2$ x 2	I	(jsr x $\pm d_7 \ldots d_2$)	$Rx \leftarrow PC; PC \leftarrow d_7 \ldots d_2$
$\pm d_7 \ldots d_2$ x 3	I	(bez x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx = 0)$
$\pm d_7 \ldots d_2$ x 4	I	(bnez x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx \neq 0)$
$\pm d_7 \ldots d_2$ x 5	I	(blz x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx < 0)$
$\pm d_7 \ldots d_2$ x 6	I	(blez x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx \leq 0)$
$\pm d_7 \ldots d_2$ x 7	I	(bgz x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx > 0)$
$\pm d_7 \ldots d_2$ x 8	I	(bgez x $\pm d_7 \ldots d_2$)	$PC \leftarrow d_7 \ldots d_2$, if $(Rx \geq 0)$

Special

Opcode	*Fmt*	*Assembly*	*Meaning*
? ? ? ? ? 0 0 0	R	(halt)	halts the machine
? ? ? ? x 1 0 0	R	(print x)	prints the contents of Rx
? ? ? ? ? 2 0 0	R	(newline)	prints a newline character