

TOY REFERENCE CARD

INSTRUCTION FORMATS

	
Format 1:	opcode d s t	(0-6, A-B)
Format 2:	opcode d addr	(7-9, C-F)

ARITHMETIC and LOGICAL operations

1: add	$R[d] \leftarrow R[s] + R[t]$
2: subtract	$R[d] \leftarrow R[s] - R[t]$
3: and	$R[d] \leftarrow R[s] \& R[t]$
4: xor	$R[d] \leftarrow R[s] \wedge R[t]$
5: shift left	$R[d] \leftarrow R[s] \ll R[t]$
6: shift right	$R[d] \leftarrow R[s] \gg R[t]$

TRANSFER between registers and memory

7: load address	$R[d] \leftarrow \text{addr}$
8: load	$R[d] \leftarrow \text{mem}[\text{addr}]$
9: store	$\text{mem}[\text{addr}] \leftarrow R[d]$
A: load indirect	$R[d] \leftarrow \text{mem}[R[t]]$
B: store indirect	$\text{mem}[R[t]] \leftarrow R[d]$

CONTROL

0: halt	halt
C: branch zero	if ($R[d] == 0$) $pc \leftarrow \text{addr}$
D: branch positive	if ($R[d] > 0$) $pc \leftarrow \text{addr}$
E: jump register	$pc \leftarrow R[d]$
F: jump and link	$R[d] \leftarrow pc; pc \leftarrow \text{addr}$

Register 0 always reads 0.
Loads from mem[FF] come from stdin.
Stores to mem[FF] go to stdout.
pc starts at 10

16-bit registers
16-bit memory locations
8-bit program counter