

Byte	0	1	2	3	4	5	6	7	8	9
halt	0	0								
nop	1	0								
cmovXX rA, rB	2	fn	rA	rB						
irmovq V, rB	3	0	F	rB						
rmmovq rA, D(rB)	4	0	rA	rB						
rrmmovq D(rB), rA	5	0	rA	rB						
OPq rA, rB	6	fn	rA	rB						
jXX Dest	7	fn								
call Dest	8	0								
ret	9	0								
pushq rA	A	0	rA	F						
popq rA	B	0	rA	F						

halt 0 0

Fetch:

icode: ifun $\leftarrow M_1[PC]$

valP $\leftarrow 1$

Decode:

int Stat:

icode == HALT: SHALT:

* cmovXX rA, rB 2 fn rA rB

Fetch:

icode: ifun $\leftarrow M_1[PC]$

rA: rB $\leftarrow M_1[PC+1]$

valP $\leftarrow PC+4$

Decode:

valA $\leftarrow R[rA]$

valB $\leftarrow 0$

Execute:

valE $\leftarrow valA + valB$

if Cond(CC, ifun), rB $\leftarrow 0xF$

Memory :

Write back :

if $\text{Cond}(\text{CC}, \text{ifun})$, $R[\text{rB}] \leftarrow \text{valE}$

PC Update :

$\text{PC} \leftarrow \text{valP}$



irmovq V, rB



Fetch :

$\text{icode} : \text{ifun} \leftarrow M_1[\text{PC}]$

$\text{rA} : \text{rB} \leftarrow M_1[\text{PC}+1]$

$\text{valC} \leftarrow M_8[\text{PC}+2]$

$\text{valP} \leftarrow \text{PC}+10$

Decode :

Execute: $\text{valE} \leftarrow 0 + \text{valC}$

Memory :

Write Back :

$R[\text{rB}] \leftarrow \text{valC}$

PC update :

$\text{PC} \leftarrow \text{valP}$



xmmovq rA, D(rB)



Fetch :

$\text{icode} : \text{ifun} \leftarrow M_1[\text{PC}]$

$\text{rA} : \text{rB} \leftarrow M_1[\text{PC}+1]$

$\text{valC} \leftarrow M_8[\text{PC}+2]$

$\text{valP} \leftarrow \text{PC}+10$

Decode :

$\text{valA} \leftarrow R[\text{rA}]$

$\text{valB} \leftarrow R[\text{rB}]$

Execute :

$\text{valE} \leftarrow \text{valB} + \text{valC}$

Memory :

$M[\text{valE}] \leftarrow \text{valA}$

Write Back :

PC update :

$\text{PC} \leftarrow \text{valP}$



mrmovq D(rB), rA



Fetch:

icode: ifun $\leftarrow M_1[PC]$

rA: rB $\leftarrow M_1[PC+1]$

valC $\leftarrow M_8[PC+2]$

valP $\leftarrow PC+10$

Decode:

valB $\leftarrow R[rB]$

Execute:

valE $\leftarrow valC + valB$

Memory:

valM $\leftarrow M_8[valE]$

Write Back:

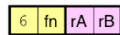
$R[rA] \leftarrow valM$

PC Update:

$PC \leftarrow valP$



OPq rA, rB



Fetch:

icode: ifun $\leftarrow M_1[PC]$

rA: rB $\leftarrow M_1[PC+1]$

valP $\leftarrow PC+2$

Execute:

valE $\leftarrow valB \text{ OP } valA$

Set CC (end, ifun)

Memory:

Write back:

$R[rB] \leftarrow valE$

PC Update

$PC \leftarrow valP$

Decode:

valA $\leftarrow R[rA]$

valB $\leftarrow R[rB]$



jXX Dest



Fetch:

icode: ifun $\leftarrow M_1[PC]$

valC $\leftarrow M_8[PC+1]$

valP $\leftarrow PC+9$

Memory:

Write Back:

PC Update:

$PC \leftarrow \text{end? } valC : valP$

Decode:

Execute:

$\text{end} \leftarrow (CC, \text{ifun})$

* call Dest

8	0	Dest
---	---	------

Fetch :

$icode \leftarrow \dots$

$valP \leftarrow PC+9$

$valC \leftarrow M_8[PC+1]$

Decode :

$valA \leftarrow R[1:rsp]$

$valB \leftarrow R[1:rsp]$

Execute :

$valE \leftarrow valA - 8$

Memory

$M[valE] \leftarrow valP$

Write Back :

$R[1:rsp] \leftarrow valE$

PC update :

$PC \leftarrow valC$

ret

9	0
---	---

F :

$icode:ifun \leftarrow M_1[PC]$

$valP \leftarrow PC+1$

D :

$valA \leftarrow R[1:rsp]$

$valB \leftarrow R[1:rsp]$

E :

$valE \leftarrow valA + 8$

M :

$valM \leftarrow M_8[valB]$

WB :

$R[1:rsp] \leftarrow valE$

PC :

$PC \leftarrow valM$

pushq rA

A	0	rA	E
---	---	----	---

F :

$icode:ifun \leftarrow M_1[PC]$

$rA:rB \leftarrow M_1[PC+1]$

$valP \leftarrow PC+2$

D :

$valA \leftarrow R[rA]$

$valB \leftarrow R[rsp]$

E : $valE \leftarrow valB - 8$

M : $M_8[valE] \leftarrow valA$

WB : $R[rsp] \leftarrow valE$

PC : $PC \leftarrow valP$

popq rA

B	0	rA	F
---	---	----	---

F:

icode := ifun $\leftarrow M_1[PC]$

rA: rB $\leftarrow M_1[PC+1]$

valP $\leftarrow PC+2$

D:

valB $\leftarrow R[rsp]$

E:

valE $\leftarrow valB+8$

M:

valM $\leftarrow M_8[valB]$

WB:

R[rsp] $\leftarrow valE$

R[rA] $\leftarrow valM$

PC:

PC $\leftarrow valP$