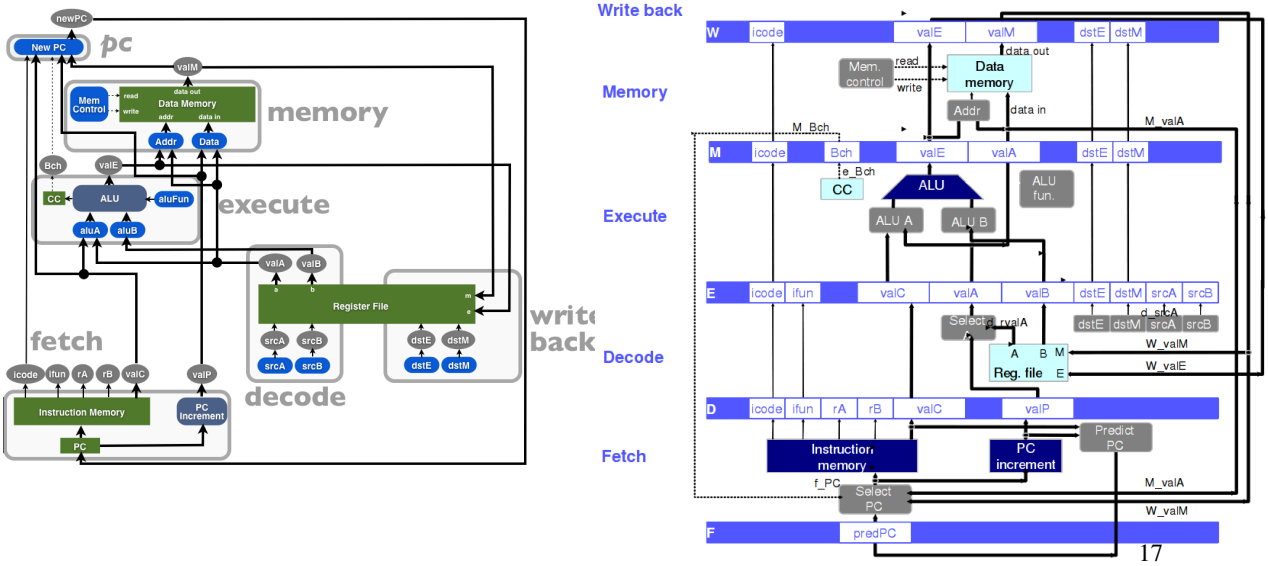
Instructions Encoding

Byte	0		1		2	3	4	5
halt	0	0						
nop	1	0						
rrmovl **rA**, **rB**	2	0	**rA**	**rB**				
cmovXX **rA**, **rB**	2	fn	**rA**	**rB**				
irmovl **V**, **rB**	3	0	F	**rB**	**V**			
rmmovl **rA**, **D**(**rB**)	4	0	**rA**	**rB**	**D**			
mrmovl **D**(**rB**), **rA**	5	0	**rA**	**rB**	**D**			
OPl **rA**, **rB**	6	fn	**rA**	**rB**				
jXX **Dest**	7	fn	**Dest**					
call **Dest**	8	0	**Dest**					
ret	9	0						
pushl **rA**	A	0	**rA**	F				
popl **rA**	B	0	**rA**	F				

Y86 ISA Reference

Instruction	Semantics	Example
rrmovl %rs, %rd	r[rd] ← r[rs]	rrmovl %eax, %ebx
irmovl $i, %rd	r[rd] ← i	irmovl $100, %eax
rmmovl %rs, D(%rd)	m[D + r[rd]] ← r[rs]	rmmovl %eax, 100(%ebx)
mrmovl D(%rs), %rd	r[rd] ← m[D + r[rs]]	mrmovl 100(%ebx), %eax
addl %rs, %rd	r[rd] ← r[rd] + r[rs]	addl %eax, %ebx
subl %rs, %rd	r[rd] ← r[rd] - r[rs]	subl %eax, %ebx
andl %rs, %rd	r[rd] ← r[rd] & r[rs]	andl %eax, %ebx
xorl %rs, %rd	r[rd] ← r[rd] $\oplus$ r[rs]	xorl %eax, %ebx
jmp D	goto D	jmp foo
jle D	goto D if last alu result ≤ 0	jle foo
jl D	goto D if last alu result < 0	jl foo
je D	goto D if last alu result $= 0$	je foo
jne D	goto D if last alu result $\neq 0$	jne foo
jge D	goto D if last alu result ≥ 0	jge foo
jg D	goto D if last alu result > 0	jg foo
call D	pushl %esp; jmp D	call foo
ret	popl %esp	ret
pushl %rs	m[r[esp] - 4] ← r[rs]; r[esp] = r[esp] - 4	pushl %eax
popl %rd	r[rd] ← m[r[esp]]; r[esp] = r[esp] + 4	popl %eax



FETCH:

```
f.iCd.set  (mem.read (f.pc.getValueProduced()+0, 1) [0] .value() >>> 4);
f.iFn.set  (mem.read (f.pc.getValueProduced()+0, 1) [0] .value() & 0xf);
f.valC.set (mem.readIntegerUnaligned (f.pc.getValueProduced()+2));
f.valP.set (f.pc.getValueProduced()+5);
```

FETCH_predict_pc:

```
f.prPc.set (f.valC.getValueProduced());
```

DECODE:

```
d.valA.set (R_NONE);
d.valB.set (R_NONE);
d.dstE.set (R_NONE);
d.dstM.set (R_NONE);
```

EXECUTE:

```
e.cnd.set (read-condition-codes-and-opcode-to-decide-if-jump-is-taken);
```

MEMORY:

WRITE BACK:

PIPELINE HAZARD CONTROL:

```
if (D.iCd.get() == I_JXX || E.iCd.get() == I_JXX ) {
    F.stall  = true;
    D.bubble = true;
}
```