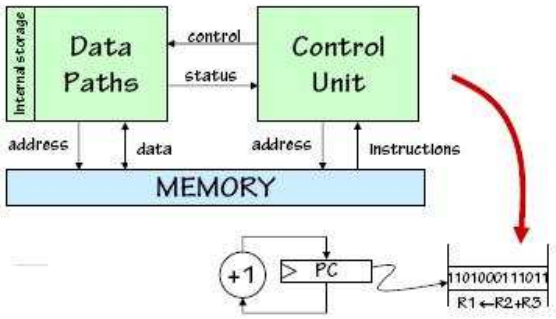
Daniel Leeds, 6.004 R12, March 24, 2006; Lecture and Tutorial Problems Excerpts

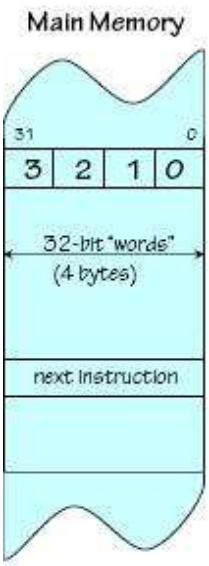
Give feedback! http://web.mit.edu/ddleeds/www/6004/form.html

von Neumann Computer
Key idea: Memory holds not only data, but *coded instructions* that make up a *program*
INSTRUCTIONS coded as binary data

PROGRAM COUNTER or **PC**:
Address of next instruction to be executed

logic to translate instructions into control signal for path

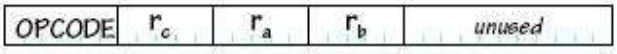


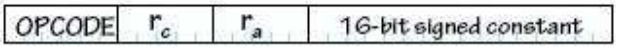


Fetch/Execute loop:

Fetch Mem[PC]
PC = PC+4
Execute fetched instruction
Repeat

Binary layout of the two Beta instruction formats:





Types of Beta assembly instructions:

Register Arithmetic:	ADD(R1,R2,R3)	R3 = R1 + R2
Memory Access:	LD(R1, c, R2)	Mem[R2+c] = R1
Compares:	CMPLT(R1,R2,R3)	R3 = 1 if R1<R2; 0 o.w.
Branch:	BEQ(R1, 0x20, R2)	R3 = PC+4 PC = PC+4+4*0x20 if R2; PC = PC+4 o.w.

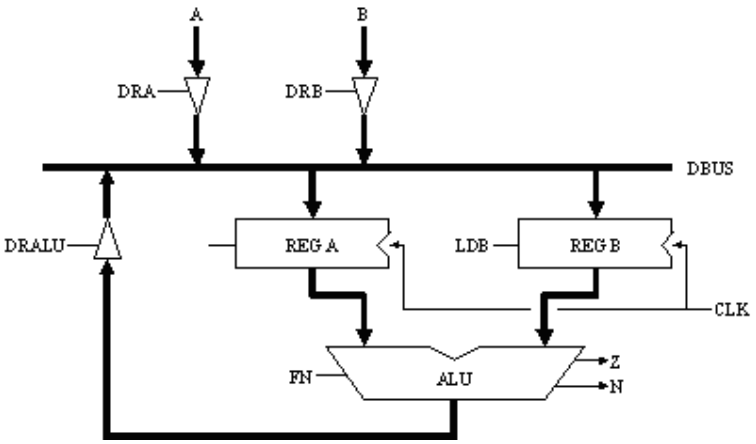
More on memory:
LD(Ra, literal, Rc): "load value at **address** *Ra+literal* into **register** *Rc*"
ST(Rc, literal, Ra): "store value in **register** *Rc* into **address** *literal+Ra*"

```
      . = 0x100
ans:  long(0)
      ...
      LD(R31, ans, R0)
```

ans = 0x100
R0 = 0 (the value at address 0x100)

Programmable Machines
Problem 1

FN=0 A-B
FN=1 B-A



F Build a controller that will cause the circuit above to execute the following algorithm:

```
while (a != b)
        if (a > b) a = a - b;
    else b = b - a;
```

	DRA	DRB	DRALU	LDA	LDB	FN
Reg A <- A	1	0	0	1	0	X
Reg B <-B	0	1	0	0	1	X
A==B?						
(if A>B) Reg A <- Reg A – Reg B						
(else) Reg B <- Reg B – Reg A						
(loop up to A=B?)						
(if A==B) do nothing						

Machine Language:
Problem 4:
D Explain why PC-relative branch addressing is a good choice for computers like the Beta that can encode only a "small" constant in each instruction.

Problem 9: Which of the following Beta instruction sequences might have resulted from compiling the following C statement?

```
int x[20], y;
y = x[1] + 4;
```

```
A LD(R31,x+4,R0)      B CMOVE(4,R0)      C LD(R31,x+4,R0)      D ADDC(R31,x+1,R0)
  ST(R0,y+4,R31)        LD(R0,x,R1)        ADDC(R0,4,R0)         ADDC(R0,4,R0)
                        ST(R1,y,R0)        ST(R0,y,R31)          ST(R0,y,R31)
```

Problem 3: A Hand assemble the following:

```
I=0x5678
B=0x1234
LD(I,R0)
SHLC(R0,2,R0)
LD(R0,B,R1)
MULC(R1,17,R1)
ST(R1,B,R0)
```

Enjoy Spring Break!