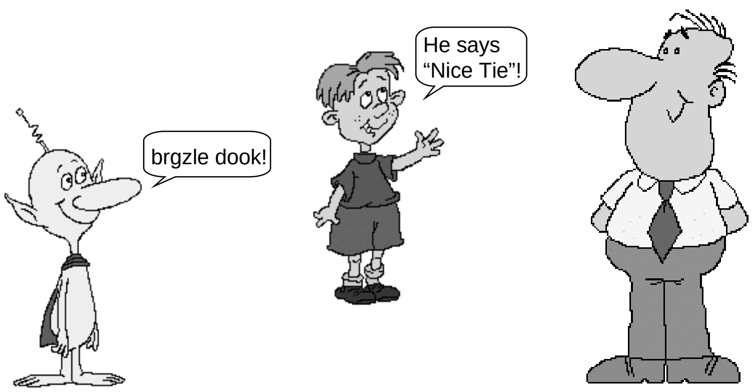
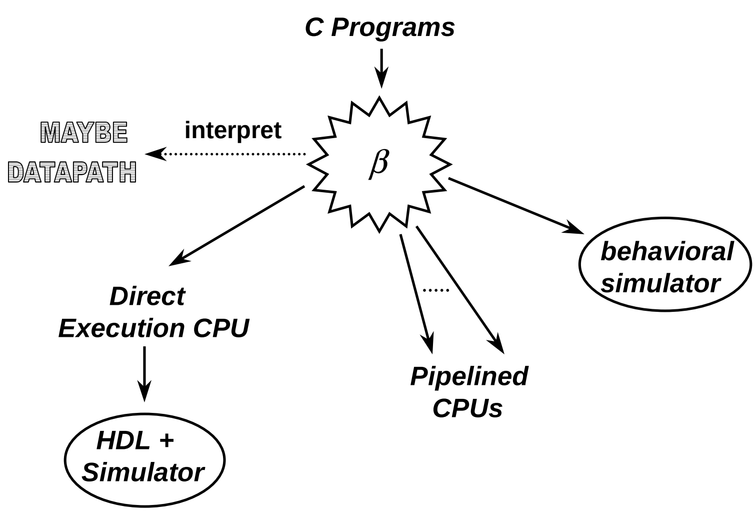
6004 Spring 1998: L28

Direct Execution
β Processor
(Can we get rid of slow interpreters?)



1

6.004 Processors and ISA’s: Big Picture



2

6004 Spring 1998: L28

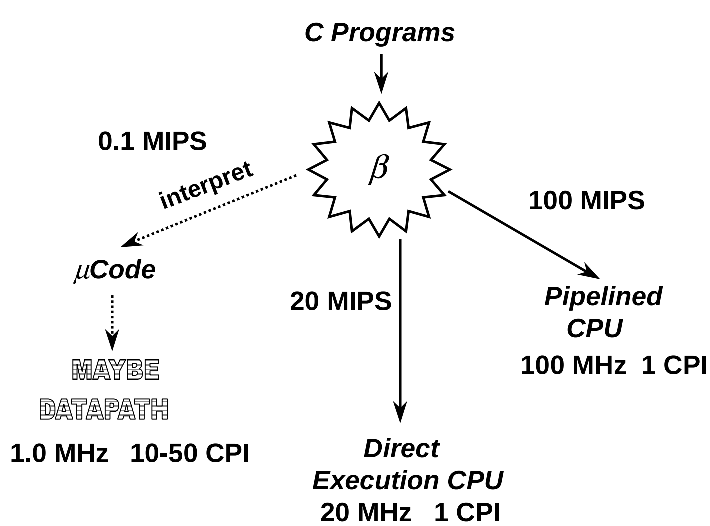
Performance Measure

$$\text{MIPS} = \frac{\text{Clock Frequency in MHz}}{\text{Clocks Per Instruction (CPI)}}$$

Millions of Instructions Per Second

3

β Performance Options



4

β Instruction Set Architecture

Form 1:

OPCODE	*rc*	*ra*	*rb*	unused
6	*5*	*5*	*5*	*11*

OPCODE = 10xxxx   *ADD:* *rc* ← *<ra>* + *<rb>*

Form 2:

OPCODE	*rc*	*ra*	literal C (signed)
6	*5*	*5*	*16*

OPCODE = 11xxxx   *ADDC:* *rc* ← *<ra>* + *C*

OPCODE = 01xxxx

LD: *rc* ← MEM[*<ra>* + *C*]
ST: MEM[*<ra>* + *C*] ← *rc*
JMP: *rc* ← *<PC> + 4; PC* ← *<ra>*
BEQ: *rc* ← *<PC> + 4; if <ra>=0 then PC*←*<PC> + 4 + C<<2*
LDR: *rc* ← MEM[*<PC> + 4 + C<<2*]

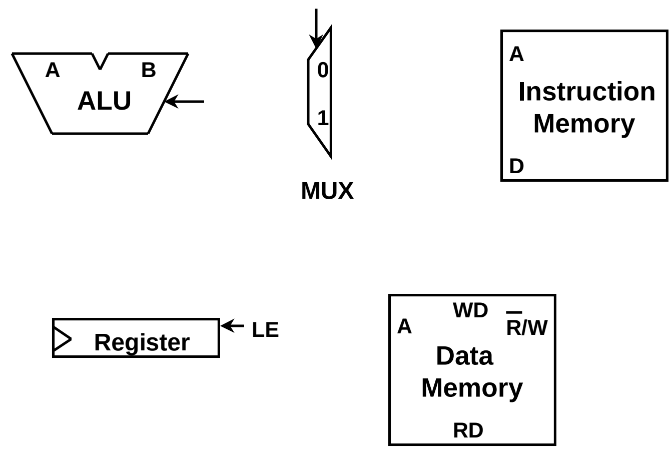
5

Approach: Incremental Design

- Divide instructions into various instruction classes based on OPCODE field
  - Operate instructions, e.g., *ADD*, *ADDC*
  - Load and Store instructions, e.g., *LD*, *ST*
  - Jump and Branch instructions, e.g., *JMP*, *BEQ*
  - Exceptions, e.g., *LDR*
- Implement datapaths for each class individually
- Merge datapaths

6

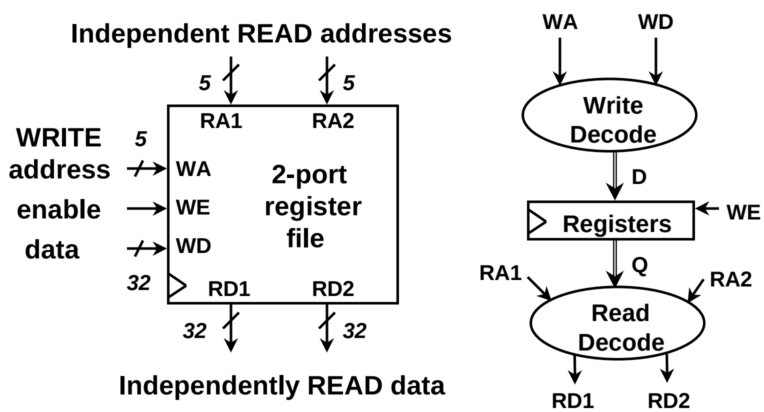
Datapath Components



7

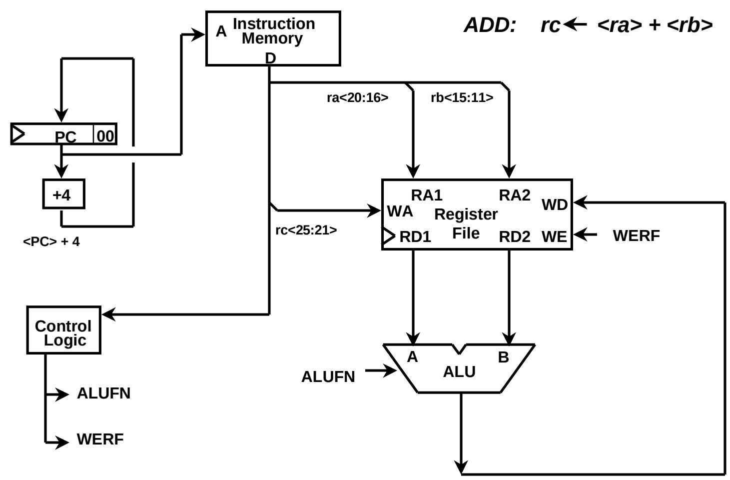
2-Port Register File

2 Combinational READ ports, 1 clocked WRITE port



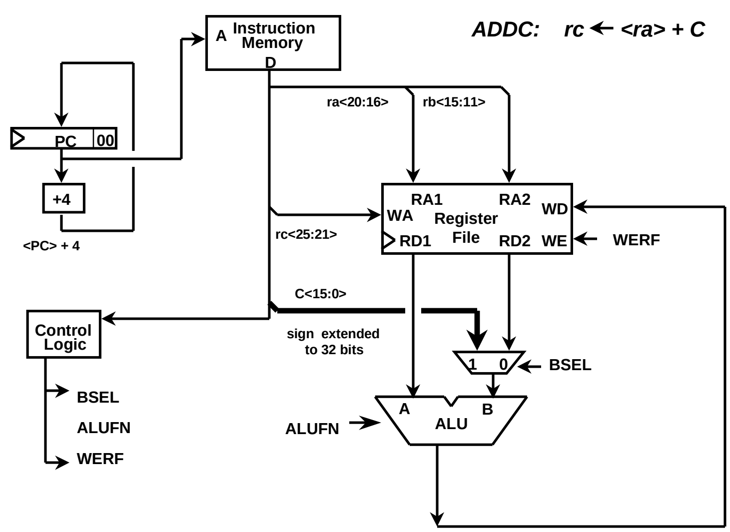
8

ALU Register-Register Operations



9

ALU Register-Constant Operations

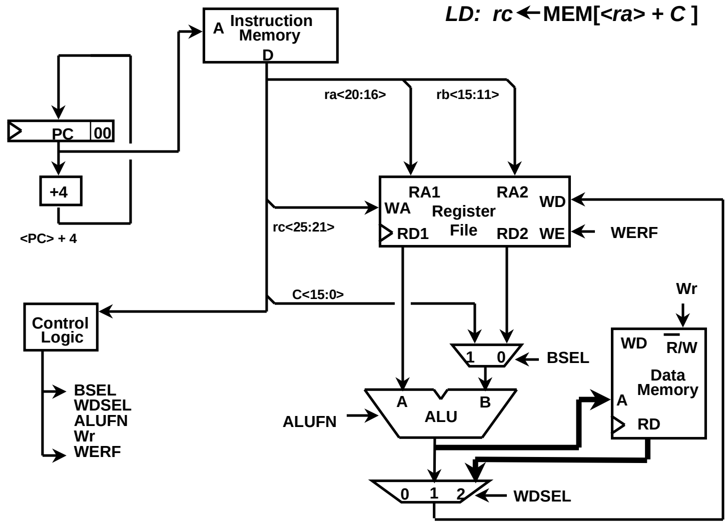


10

6004 Spring 1998: L28

Load Instruction

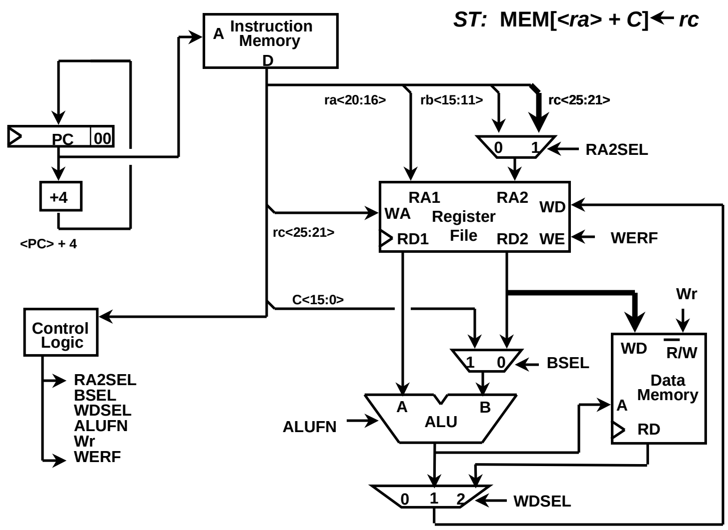
LD: rc ← MEM[<*ra*> + *C*]



11

Store Instruction

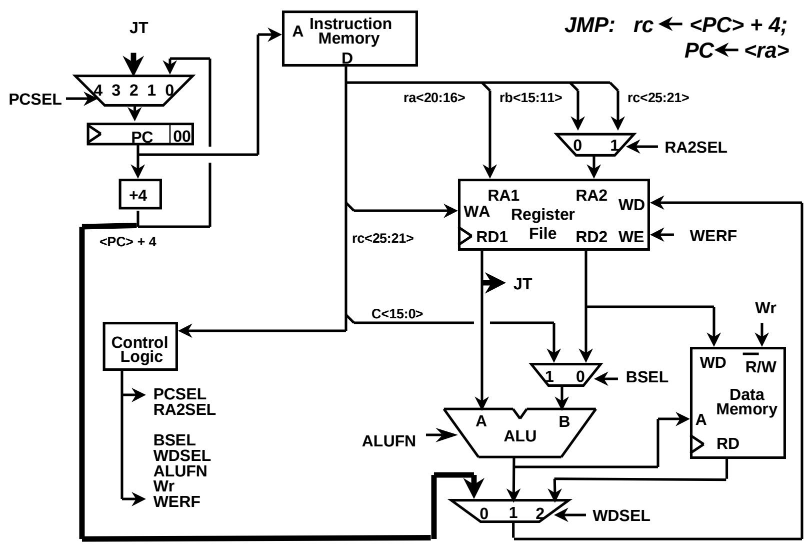
ST: MEM[<*ra*> + *C*]← *rc*



12

Jump Instruction

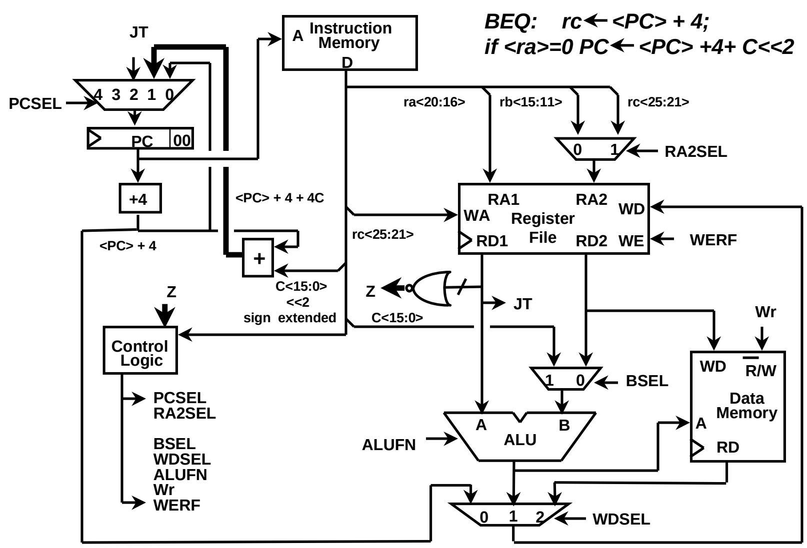
JMP: rc ← <PC> + 4;
PC ← <ra>



13

Branch Instructions

BEQ: rc ← <PC> + 4;
if <ra>=0 PC ← <PC> +4+ C<<2

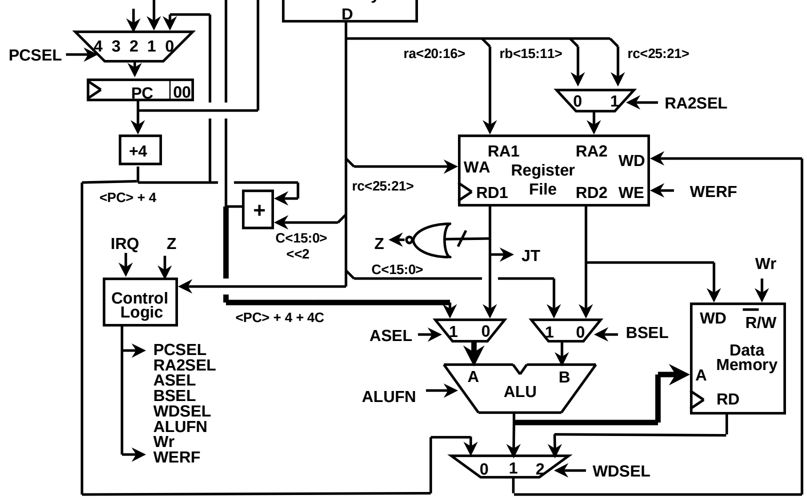


14

6004 Spring 1998: L28

Load Relative Instruction

LDR: rc ← MEM[<PC> +4+ C<<2]



15

What's LDR Good For Anyway?

*X = X * 123456*

LD(r31, X, r0)
LDR(c1, r1)
MUL(r0, r1, r0)
ST(r0, X, r31)

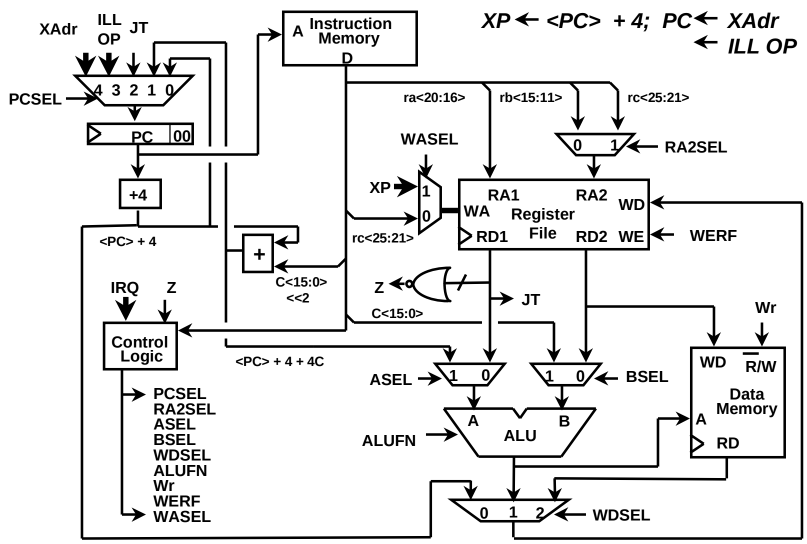
c1: long(123456)

Why can't we use MULC(r0, 123456, r0) ?

16

6004 Spring 1998: L28

Exceptions: Traps, Faults and Interrupts



17

Specifying Combinational Control Logic

	INSTR.	*OP*	*OPC*	*LD*	*ST*	*BEQ*	*BNE*	*JMP*	*...*
Control logic inputs	OPCODE								
	Z								
	IRQ								
Control logic outputs	PCSEL								
	RA2SEL								
	ASEL								
	BSEL								
	WDSEL								
	ALUFN								
	Wr								
	WERF								
	WASEL								

Looking at the datapath you should be able to fill in the entire table!

18