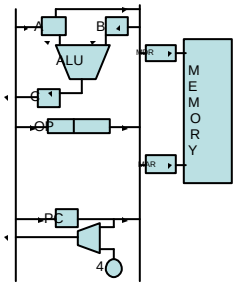
Computer Architecture

Lecture 7

1 / 21 / 2004

2-BUS Architecture



Instruction Formats

- The Format of the current instructions are OP <ADDR>
- To provide instructions of the format

  ADD, R1, R2, R3

  and

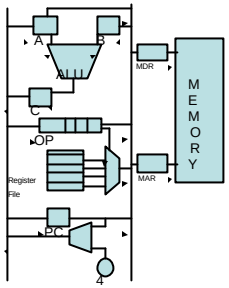
  STORE R3, <ADDR>

  where R1, R2 and R3 are registers we need to add an architecture to select the appropriate register

Register File

- By utilizing a register file you can organize the registers in one array
- The control unit allows us to select the appropriate register for the operation and load it into its target location (either memory or another register)
- This is a crucial aspect of the Load/Store architecture

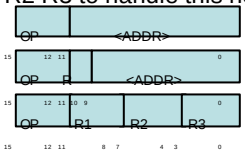
2-BUS Load/Store Architecture with Register File



Instruction Formats

- The format of the instruction word must also be altered
- The format OP <ADDR> needs to be changed to OP R <ADDR> and

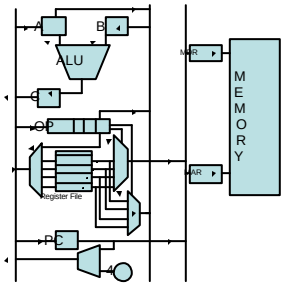
  OP R1 R2 R3 to handle this new design



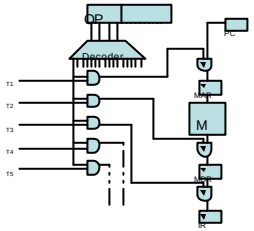
3-BUS Load/Store Architecture with Register File

- By adding a third BUS the system can load two registers at once using each BUS
- Two multiplexers need to be attached to the output of the register file: one connected to each BUS to send the contents of the target register to its destination
- Another multiplexer unit is added to send input from the third BUS into the correct location within the register file

3-BUS Load/Store Architecture with Register File

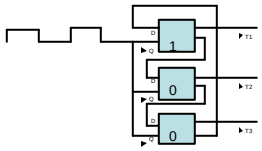


Control Unit



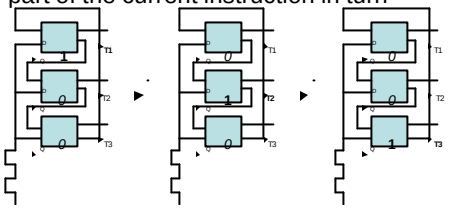
Clock Unit

- The Latch will only allow the change of the value on each clock tick
- By linking together they simulate a counter that will only allow one operation in a single clock tick

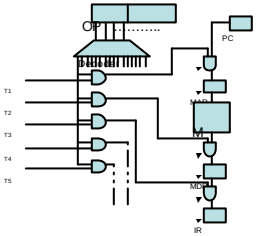


Clock Unit

- At every clock tick the active flip-flop changes, sending voltage to each different part of the current instruction in turn

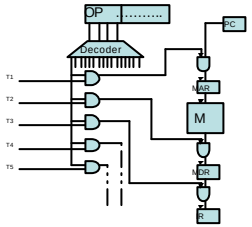


Control Unit



Each line emerging from the decoder represents an operation. When the decoder is set to that operation, it sends voltage down the appropriate channel which is ANDed with the signals coming from the clock. This allows for precise timing in the step executions of instructions and makes synchronization among components possible.

Control Unit

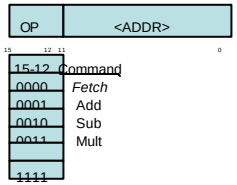


Fetch Cycle:
T1: MAR <- PC
T2: MDR <- M[MAR]
T3: IR <- MDR
T4: MAR <- IR.ADDR
T5: Decoder <- IR.OP
The chain of flip-flops will be as long in clock ticks as the ongest command takes to execute.

One-Address Format

- The operation is 4 bits and the remaining 12 are address bits
- This gives the availability of 2^4 operations and 2^{12} memory locations

One-Address Format



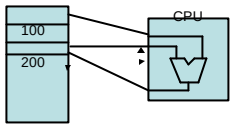
15-12	Command
0000	*Fetch*
0001	Add
0010	Sub
0011	Mult
1111	

- 0000 is the Fetch operation
- It is a hidden instruction that cannot be accessed by the user

Two-Address Format

- The operation is performed on the memory addresses of the first and second operands and the value is stored back in the location specified by the first operand field

Two-Address Format



This is analogous to:

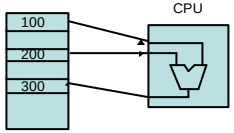
Load 200
Add 100
Store 200

all performed in one command

Three-Address Format

- The operation is performed on the memory addresses of the first and second operands and the value is stored in the location specified by the result address field
- This format is not often used because it requires three memory accesses per operation (which is very slow)

Three-Address Format



This is analogous to:

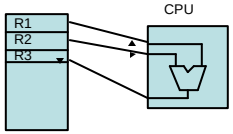
Load 200

Add 100

Store 300

all performed in one command

Another Three-Address Format



In a Load/Store machine, the addresses are often locations in the register file rather than in memory. This is very fast on a machine with multiple BUS architecture.