Basic Processing Unit

Three main parts of the CPU:

1. Control Unit
2. ALU
3. Register file
4. Processor memory interface
5. instruction address generator (with PC and IR)

0.1 Datapath

The datapath: It is a collection of Functional units, registers, memory interfaces that perform every instruction in the ISA

- The pipeline is distributed into 5 stages along the datapath
  1. Fetch: Get the instruction pointed to by the PC, store it in IR
     Memory address ← [PC], Read memory, Wait for MFC, IR ← Memory data, PC ←[PC] + 4
  2. Decode: The instruction is decoded and the source registers are read
     RA ← [Address A], RB ← [Address B], Op Code ← [depending on instruction]
     Note all these addresses locations are dependent on the type of instruction as below
  3. Execute: The computation specific for the instruction is performed. RZ ← computation result, RM ← RB
     An immediate value is selected from MuxB if the need be. RM is made RB if the instruction is one involving memory read or write (in that case RZ hold the effective address)
  4. Memory: If instruction is a memory instruction then perform it in this stage.
     If not memory: RY ← RZ
     If Load: RY ← memory data
     If Store: RY ← RZ (for writeback just in case)
     If Branch: RY ← Return address [to be written to link register]
  5. Write-Back: The result of the instructions operation is stored in the destination register. Address of C determined by corresponding IR bits
     addressC ← RY
- There are multiple inter-stage buffers that pass information along
- There are also multiple control signals responsible for controlling the flow of information.
  1. generate individual signals per step of instruction in the data-path
  2. control memory access (read/write)
  3. determine which registers to be accessed/written to
  4. determine MUX control signals
  5. determine branch flow by controlling PC signals
  6. determine operation for the ALU
  7. determine the type of the instruction:
     (a) Register Operand format

$$R_{src1}(31-27)|R_{src2}(26-22)|R_{dst}(21-17)|OP\ code(16-0)$$

     (b) Immediate Operand format

$$R_{src1}(31-27)|R_{dst}(26-22)|Immediate\ operand(21-6)|OPcode(5-0)$$

     (c) Call format

$$Immediate\ operand(31-6)|OPcode(5-0)$$

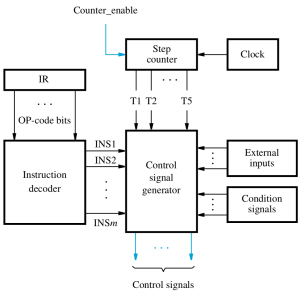
0.2 Control Signals

The control circuitry: implements the decode stage of the instruction execution, and sets the appropriate signals at each stage to regulate the flow of data.

- Register File
  1. RF_write: When writing to register file it is set to 1
  2. C_select: Depending on instruction format
     - R-type: dest is IR_{21-17}
     - I-type: dest is IR_{26-22}
     - Branch: dest is `LINK`
- ALU
  1. B_select: 0 for RB, 1 for immediate
  2. ALU_op: k-bit control signals (for each type of operation)
  3. Condition signals: Output of ALU, set by the ALU, and monitored for branch conditions
- Memory and IR:
  1. MEM_read: Set when reading
  2. MEM_write: Set when writing
  3. IR_enable: loads intruction into IR, and activated in Fetch stage after MFC is asserted
  4. MA_select: choose between RZ and PC for choosing the address input to memory
  5. MFC: stall signal, if cache hit then asserted in same clock cyle as request, otherwise instruction stalled until memory access complete
- Instruction Address Generator:
  1. INC_select: Chooses between +4 and branch offset for PC incrementation
  2. PC_select: Chooses between subroutine RA(0) and incremented PC(1), and needs PC_enable to be on

There are 2 ways to generate these control signals:

1. Hardwired control: Decoder interprets Op-code and address mode in IR, while step counter (modulo 5) sets T# to indicate which stage to execute, while control signal generator is a combinational circuit responsible for generating the right signals



2. Microprogrammed control: Using software (not in TB or notes)