2017-10-31

- For every instruction, the first two steps are identical:
  - Send the **program counter (PC)** to the memory that contains the code and fetch the instruction from that memory.
  - Read one or two registers, using fields of the instruction to select the registers to read.
- **Multiplexor** or **data selector**: a logic element that chooses from among the multiple sources and steers one of those sources to its destination.
- **Control unit**: has the instruction as an input, is used to determine how to set the control lines for the functional units and two of the multiplexors.
- The datapath elements in the MIPS implementation consist of two different types of logic elements:
  - **Combinational element**: elements that operate on data values.
  - **State elements**: elements that contain state, e.g. *memories* and *registers*.
- A state element has at least two inputs and one output.
  - The required inputs are the data value to be written into the element and the *clock*.
  - The output from a state element provides the value that was written in an earlier clock cycle.
- The **clock** is used to determine when the state element should be written; a state element can be read at any time.
- **Edge-triggered clocking**: any values stored in a sequential logic element are updated only on a clock edge.
- Because only state elements can store a data value, any collection of combinational logic must have its inputs come from a set of state elements and its outputs written into a set of state elements.
- **Control signal**: a signal used for multiplexor selection or for directing the operation of a functional unit.
- Both the clock signal and the write control signal are inputs, and the state element is changed only when the write control signal is *asserted* and a clock edge occurs.
- **Datapath element**: a unit used to operate on or hold data within a processor.
- A **register file** is a collection of registers in which any register can be read or written by specifying the number of the register in the file.
- **Branch target address** is computed by adding the sign-extended offset field of the instruction to the PC.
- The jump instruction operates by replacing the lower 28 bits of the PC with the lower 26 bits of the instruction shifted left by 2 bits.

- The MIPS ALU in defines the 6 following combinations of 4-bit control inputs: and, or, add, subtract, set on less than, nor.
- 4-bit ALU control input is generated using a small control unit that has as inputs the function field of the instruction and a 2-bit control field (ALUOp):
  - add (ALUOp = 00) for loads and stores.
  - subtract (ALUOp = 01) for *beq*.
  - determined by the operation encoded in the *funct* field (ALUOp = 10).

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111

- As a step in designing this logic, it is useful to create a **truth table** for the interesting combinations of the function code field and the ALUOp bits.
- The effect of each of the seven control signals:

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20:16).	The register destination number for the Write register comes from the rd field (bits 15:11).
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, lower 16 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

- The control line *PCSrc* should be asserted if the instruction is *branch on equal* (a decision that the control unit can make) and the *zero* output of the ALU.
- For a control function, the outputs are the control lines, and the input is the 6-bit opcode field, *Op [5:0]*.
- **Single-cycle implementation**: an implementation in which an instruction is executed in one clock cycle.
- We can implement a jump by storing into the PC the concatenation of (1) the upper 4 bits of the

current PC + 4, (2) the 26-bit immediate field of the jump instruction, and (3) the bits 00.

- Although the single-cycle design will work correctly, it would not be used in modern designs because it is inefficient since the clock cycle is too long.
- The clock cycle is equal to the worst-case delay for all instructions, so it's useless to try implementation techniques that reduce the delay of the common case but do not improve the worst-case cycle time.