Laura Gretz
Kyle Eaton
Fabienne Guertler
Series 4 - Computer Architecture

Sign Extension

a. The 16-bit offset is either a positive or a negative number once its converted to 32 bits it needs to keep the sign information by copying the most significant bit to the upper 16 bits

Logical and Bitwise Operations

b. 22
c. 0
d. BB
e. 0xEF
f. 1
g. 0

Hardware Access

a. A processor could get access to connected hardware when all the gates are always working
b. With interfaces where the hardware provides relevant information.

bne instead of beq

a. Singlecycle implementation: put a NOT gate at the 0 output of the ALU
b. Multicycle implementation: analog to singlecycle implementation.

Pipeline Registers

These state registers are added between each pipeline stage to isolate them:

a. PC is the pipeline register that feeds the IF stage of the pipeline
b. IF: Instruction Fetch -- Fetched instruction from memory and places it in an instruction register
c. ID: Instruction Decode -- instructions are decoded and figures out what the instruction says to do and grabs the values from names variables
d. EX:Execute -- instruction is executed using the ALU
e. MEM:Memory Access -- if the instruction is a load or store the memory will be either read or written
f. WB:Write Back -- the results of the instruction or operation are written to the explicit register

Pipelining Hazard

a. Control Hazards: can occur in pipelining branches that change the program counter and makes a decision about a program control flow before a condition has been evaluated
b. Data Hazards: Register usage can cause data hazards (i.e. read before write), they happen when an instruction depends on the result of a previous instruction and it tries to use the data before it is ready.
c. Structural Hazards: if there is only one memory it can try to use the same resource by two different instructions happening at the same time.

Stall

a. On slide 19 the results of the 4th cycle of the *beq* instruction are used in the first cycle of the *lw* instruction. On slide 15 it uses the results of the 2nd cycle.

Data Hazard

add $t0, $t5, $t4	
lw $s2, 0($t0)	forwarding
sub $s3, $t0 ,$s2	stall

sw $t4, 4($s3) forwarding

Forwarding doesn’t help if an instruction tries to read a register following a load instruction that writes to the same register.