John Carbeck
16309095
CS2022
Computer Architecture

Simulation Discussion

ALU:

Signals are Cin (carry in), A (input 1), B (input 2), S0, S1 and S2 (operation selection), G (result), and Cout (carry out). The simulation shown is for a 1 bit ALU. The simulation correctly shows the operation of a 1 bit ALU where the output G is at a 5ns delay.

ALU 16 bit:

Here the correct operation is shown for a 16 bit ALU. The signals are the Gsel vector, A (16 bit value 1), B (16 bit value 2), The flags V, C, N and V, and the ALU output G. The simulation here shows all the possible operations capable by the ALU with values that demonstrate the appropriate values for flags based off input and results, based off given table of operations.

Arithmetic Circuit:

Here is a 1 bit arithmetic circuit. The signals are A (1 bit value 1), B (1 bit value 2), Cin (carry in), S0 and S1 (operation select), Z (output 1 bit), and Cout (the carry out bit). This simulation shows the correct operation of a arithmetic circuit, which includes the correct output and carry out based of the given inputs, based off given table of operations.

CAR:

Here is a simulation for a the control address register. The signals shown here are A(load in the input to output), clk (clock), reset, and Z (output to control memory). The input of the reset signal shows the access of the address 0xC0 which contains the fetch instruction from memory location. This simulation shows the input will only be shown on the output when the signal A is high. This is the correct operation for this entity.

Control Memory:

This simulation shows how the control memory accesses demultiplex to the correct output the correct output signals. The IN_CAR is the address of memory that should be accessed. The signals that are resulting is based off the following values:

```
0: x"C020306"
1: x"C02400E"
2: x"C020184"
3: x"FFFFFFF"
```

(the shown simulation is correct)

Datapath:

This simulation shows the correct operation of the datapath. Here the values are loaded into registers. Through the various control signals all operations in the given FS table are show. These operations results are loaded into R0. The clock cycle is at 50ns. This is to make sure that the output from the functional unit and their flags are appropriately set. The flag values are dependent on the output of the functional unit.

Decoder 4 to 9:

Here the results show the correct operation of a 4 bit to 9 output decoder. This is used in the selection of the temp register in the register file. Here all 9 outputs are accessed based off a 4 bit value. The selection signal is AND with the RW signal. This allows for the control signals to dictate if the value on the data path is loaded into the specified register only when RW is set. The signals here are RW (write), A0-A3 (selection signals), and Q0-Q8 (output selection signals).

Extend:

Here the results show the correct performance of the extend circuit for the program counter. This takes in the SA and DR from the instruction register and takes the most significant bit [5] and if it is a 1, ie. two's complement negative, the more significant bits also become 1. Transforming the 6 bit value to a 16 bit one, while keeping the sign of the number. The signals here are DRSA(6 bit output from instruction reg) and Ext (the signed 16 bit output).

Full Adder:

This simulation of a full adder shows the correct behaviour of a 1 bit adder. This includes the addition carry in signal. The signals here are A and B (1 bit inputs), Cin (1 bit carry in), Cout (1 bit carry out), and Z (1 bit output).

Functional Unit:

This simulation shows the correct operation for all the FS signals. The signals here are A and B (16 bit inputs), F (16 bit output), FS (function selection), and the flags V, C, N, and Z. The values used as inputs were selected to also show the correct flag output. The clock speed chosen for this simulation was 50ns. This was to make sure the output of the functional unit, the result and flags, were set correctly.

Instruction Register:

This simulation shows the correct behaviour of a instruction register. As shown in the simulation the input is only shown on the output if the instruction load signal is high. The values that are broken down to the opcode, DR, SA, and SB signals are appropriate. The entity is also clocked. The signals are IRin (instruction reg in), IL (instruction load), clk (clock), opcode (control memory address for operation), DR (data load select), SA (ABUS select), and SB (BBUS select).

Logic Unit:

This simulation shows the correct behaviour of a 1 bit logic unit. Showing the outputs AND, OR, XOR, and NOT. The signals are A and B (input), S0 and S1 (operation select), and G (output).

LR 16 bit Shifter:

This simulation shows the correct behaviour of a 16 bit LR shifter. The signals here are B (16 bit input), Hsel (the operation select), and H (16 bit output).

Memory:

This simulation shows the correct operation of memory. Here the first value in memory is read, written to, and then read again to determine if the change occured. Here the signals are address (16 bit memory address), DataIn (16 bit data value to be written to memory), clk (clock), MW (the signal that allows for memory to be written), and DataOut (the value accessed by the address given).

Micro control:

This simulation shows the correct operation of the control aspect of the processor. Here it is shown that all elements that make up the controller of the datapath and memory use the appropriate signals when given certain instructions. This also shows that the signals from control memory effect the other entities that use them in the micro control entity.

Mux 2 to 1:

Here the simulation shows the correct operation of a 2 to 1 mux where the output of the the mux is determined by a low or high selection signal.

Mux 2 to 8:

Here the simulation shows the correct operation of a 2 to 8 mux where two 8 bit values are the 8 bit output depending on a low or high selection signal.

Mux 2 to 16:

Here the simulation shows the correct operation of a 2 to 16 mux where the two 16 bit values are set to the output depending on the low or high selection signal.

Mux 3 to 1:

Here the simulation shows the correct operation of a 3 to 1 mux where three 1 bit values are chosen as output depending on the combination of values from the two select signals.

Mux 8 to 1:

Here the simulation shows the correct operation of a 8 to 1 mux where eight 1 bit values are chosen as output depending on the combination of values from the three select signals.

Mux 8 to 16:

Here the simulation shows the correct operation of a 8 to 16 mux where eight 16 bit values are chosen as a 16 bit output depending on the combination of values from the three select signals.

Mux 9 to 16:

Here the simulation shows the correct operation of a 9 to 16 mux where nine 16 bit values are chosen as a 16 bit output depending on the combination of values from the four select signals.

Programme Counter:

Here the simulation shows the correct operation of a programme counter. The value in the PC is incremented when PI is set high, the value in the PCin is loaded into PC when PL is high, the PC is loaded by the next value when PI and PL are set. Reset set the programme counter back to 0. These operations are also synced with the clock.

Programme CPU:

Here the simulation is showing the correct operation of the Programme CPU which contains the all the entities. The operations that are performed are the ones that have been written in memory and dictated in the control memory. (note not all the instructions were finished in time).

Register:

Here the simulation is showing the loading of values into a 16 bit register when both RW are high and the clk is high. The value that is stored in the register is shown in the signal Q.

Register File:

Here the simulation is showing the loading of values into all the registers including the temp. There is also a register transfer that occurs. This is the correct operation for the register file.

Zero Detect:

Here the simulation is showing the correct behaviour of a zero detect, the value Z is high when the value it's checking is all zeros.

ZeroFill:

Here the simulation is showing the correct behaviour of a zero fill where a smaller 3 bit value is being extended with zeros to be the same value but at a larger 16 bit value