When RW = 1 and MD = 1:

|  |  |  |
| --- | --- | --- |
| Inputs | | Outputs |
| DATA\_IN | DR(Values 00000 -> 11111)  TD = 0000 | Value in DATA\_IN written into Register 0-31 |

|  |  |  |
| --- | --- | --- |
| Inputs | | Outputs |
| DATA\_IN | TD(Values 0001 -> 1111) | Value in DATA\_IN written into TempReg 1-15 |

In the Datapath, values are written into one of the 32 registers or one of the 15 temp registers by taking in a 32-bit value from DATA\_IN. This value is inputted into MuxD, which determines if the value inputted into the register file is DATA\_IN or the output of the functional unit. When the signal MD is 1, DATA\_IN is selected.

Inside the register file there are two multiplexers, one for determining which of the main registers is written into and the other for determining which of the temp registers is written into. This is determined by two signals: DR, the 5-bit signal which selects a main register, and TD, the 4-bit signal which selects a temp register. A main register is only selected if TD = 0000. A value is only written into a register if the signal RW is 1, in which case the 32-bit value of DATA\_IN is written into a register.

In the simulation attached, the binary value of my student number, 22335824, is written into register 0, and for every subsequent register and temp register I subtracted 1 from the previous register and wrote that value into the register.

For A = 22335815 (register 9), B = 22335805 (register 19) and the destination register = register 4:

|  |  |  |  |
| --- | --- | --- | --- |
| Cases | FS | Operation | Value in register 4 |
| A | 00101 | A – B | 00000000000000000000000000001010 (10) |
| B | 01110 | NOT A | 11111110101010110010111010111000 (-22335816) |
| C | 00100 | A + 1's c B | 00000000000000000000000000001001 (9) |
| D | 00111 | A | 00000001010101001101000101000111 (22335815) |
| E | 00011 | A + B + 1 | 00000010101010011010001010000101 (44671621) |
| F | 10100 | srB | 00000000101010100110100010011110 (11167902) |
| G | 00010 | A + B | 00000010101010011010001010000100 (44671620) |
| H | 11000 | slB | 00000010101010011010001001111010 (44671610) |
| I | 00000 | A | 00000001010101001101000101000111 (22335815) |
| J | 10000 | B | 00000001010101001101000100111101 (22335805) |
| K | 00001 | A + 1 | 00000001010101001101000101001000 (22335816) |
| L | 01100 | A XOR B | 00000000000000000000000001111010 (122) |
| M | 01010 | A OR B | 00000001010101001101000101111111 (22335871) |
| N | 00110 | A – 1 | 00000001010101001101000101000110 (22335814) |
| O | 01000 | A AND B | 00000001010101001101000100000101 (22335749) |

Using the SA and SB, two 5-bit values, you can choose which registers are read from ports A and B. Port A goes directly to the Functional Unit, while Port B goes to a 2 to 32 multiplexer, MuxB. With this, the second input into the Functional Unit can be either the value of B or a constant 32-bit value IR\_IN, depending on the value of the signal MB. When it is 0, B is inputted into the Functional Unit.

Here, logical operations, additions and subtractions are performed on the inputs, depending on the value of the 5-bit FS input, as demonstrated in the above table. The output of the Functional Unit is then connected to MuxD, which inputs the output of the Functional Unit into the Register File when MD = 0. The value of the output of the Functional Unit is then written into a selected register when RW = 1.

The following is the operations performed involving B when using my student number as a constant input from IR\_IN.

When A = 22335815 (register 9), B = 22335824 (from IR\_IN), and the destination register = register 4:

|  |  |  |  |
| --- | --- | --- | --- |
| Cases | FS | Operation | Value in register 4 |
| P | 00101 | A-B | 11111111111111111111111111110111 (-9) |
| Q | 00100 | A + 1's c B | 11111111111111111111111111110110 (-10) |
| R | 00011 | A+B+1 | 00000010101010011010001010011000 (44671640) |
| S | 10100 | srB | 00000000101010100110100010101000 (11167912) |
| T | 00010 | A+B | 00000010101010011010001010010111 (44671639) |
| U | 11000 | slB | 00000010101010011010001010100000 (44671648) |
| V | 10000 | B | 00000001010101001101000101010000 (22335824) |
| W | 01100 | A XOR B | 00000000000000000000000000010111 (23) |
| X | 01010 | A OR B | 00000001010101001101000101010111 (22335831) |
| Y | 01000 | A AND B | 00000001010101001101000101000000 (22335808) |