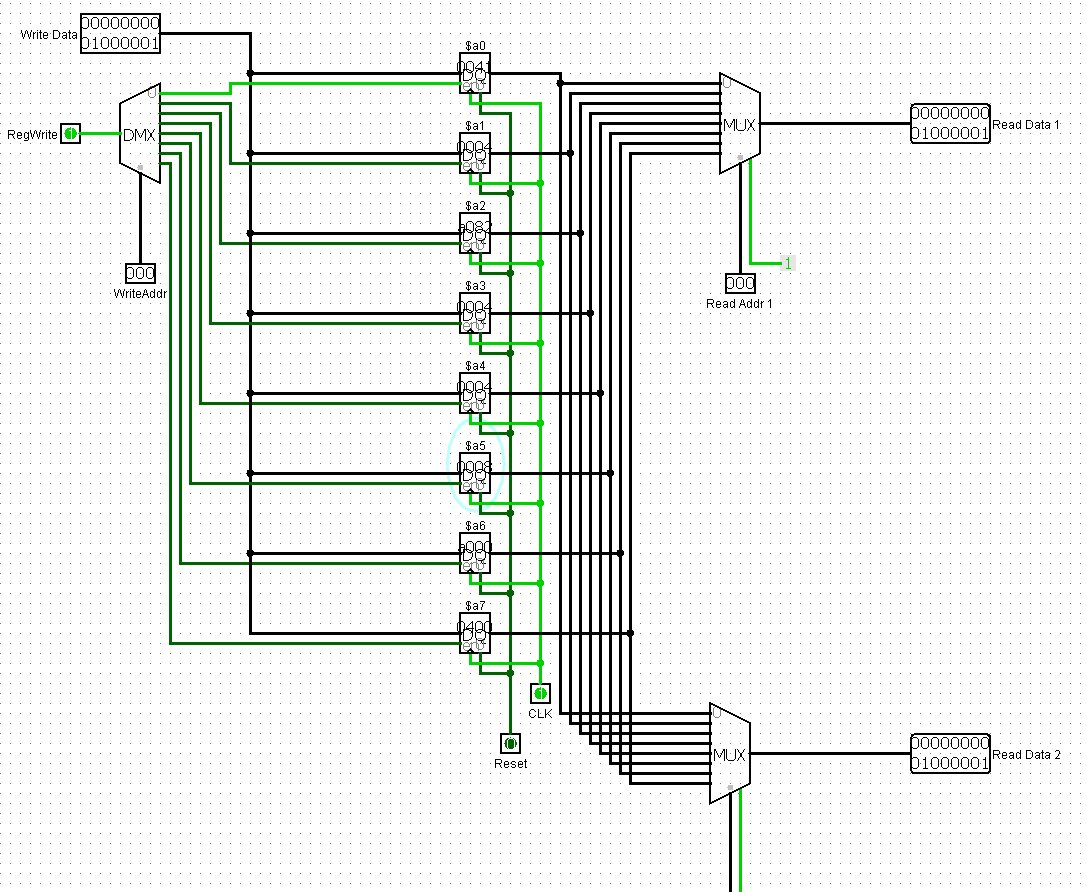
Tracing the Test Program Vector-Vector Multiplication

First we loaded code file to instruction memory and data file to data memory. Before starting the program, we wrote $4 0 and $5 4 as the base addresses of arrays a and b in our register file respectively. In the first two clock cycles, we wrote $1 and $2 zero. In the third clock cycle, $3 is written 4 because of addi $3, $0, 4 instruction.

Then our program goes into loop. In the loop, in the fifth clock cycle $6 becomes 0005 since the first value in array a. In the sixth clock cycle $7 becomes 0041 (which is 65 in decimal) which is the first element of second array b. Then we multiply and then load the low of multiplication to $6. In the eight clock cycle, $6 becomes 145 (which is 5\*65= 325 in decimal). Then we increase $1 (which is i) and base addresses for load operations in the loop and jump back to the beginning of loop again and we saw that the address in the instruction memory is jumping again to memory address 3 which beq, beginning of loop. After jump instruction PC came back to the beginning of loop address and we do again branch instruction.



This is the last image of our instruction memory after executing all instruction. So in the end, the result “sum” is in the $a2 as we see it is now a082 in hex (which is 41090 in decimal) the result of the vector-vector multiplication.

And also in the end we executed our new instruction we created, So it loaded 0041 to $a0 as we wanted.