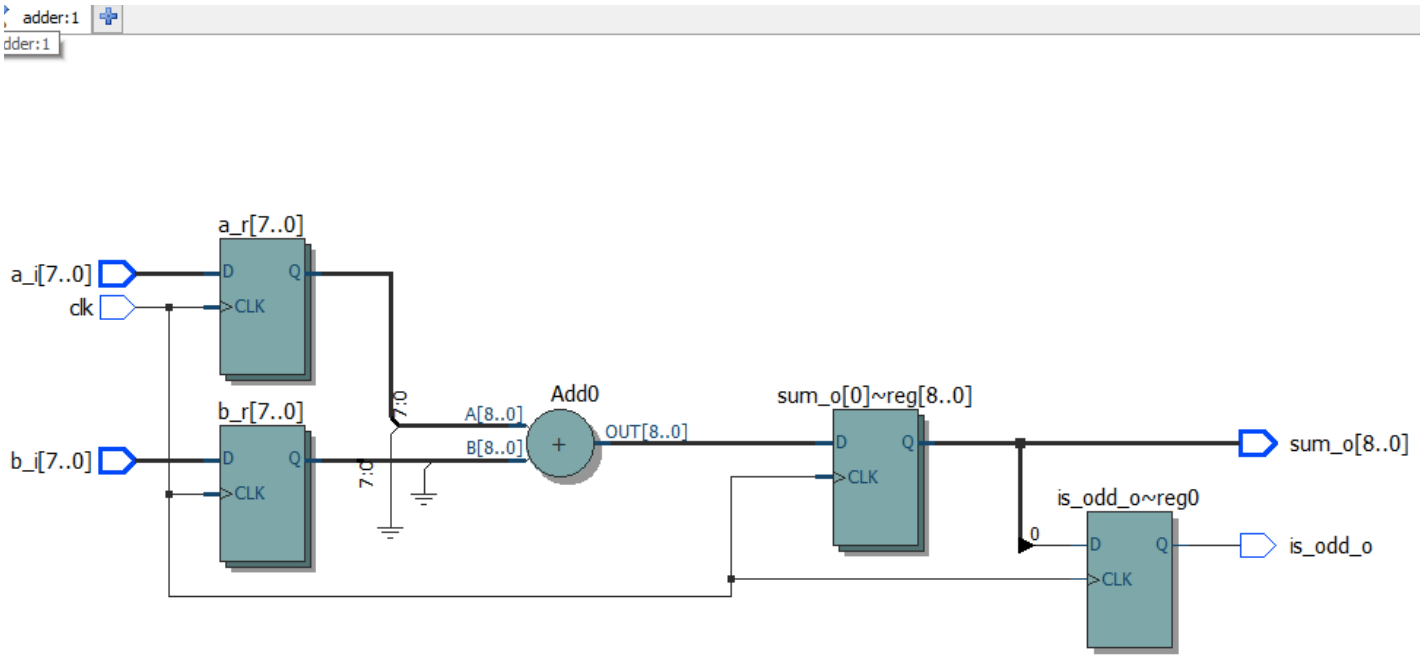
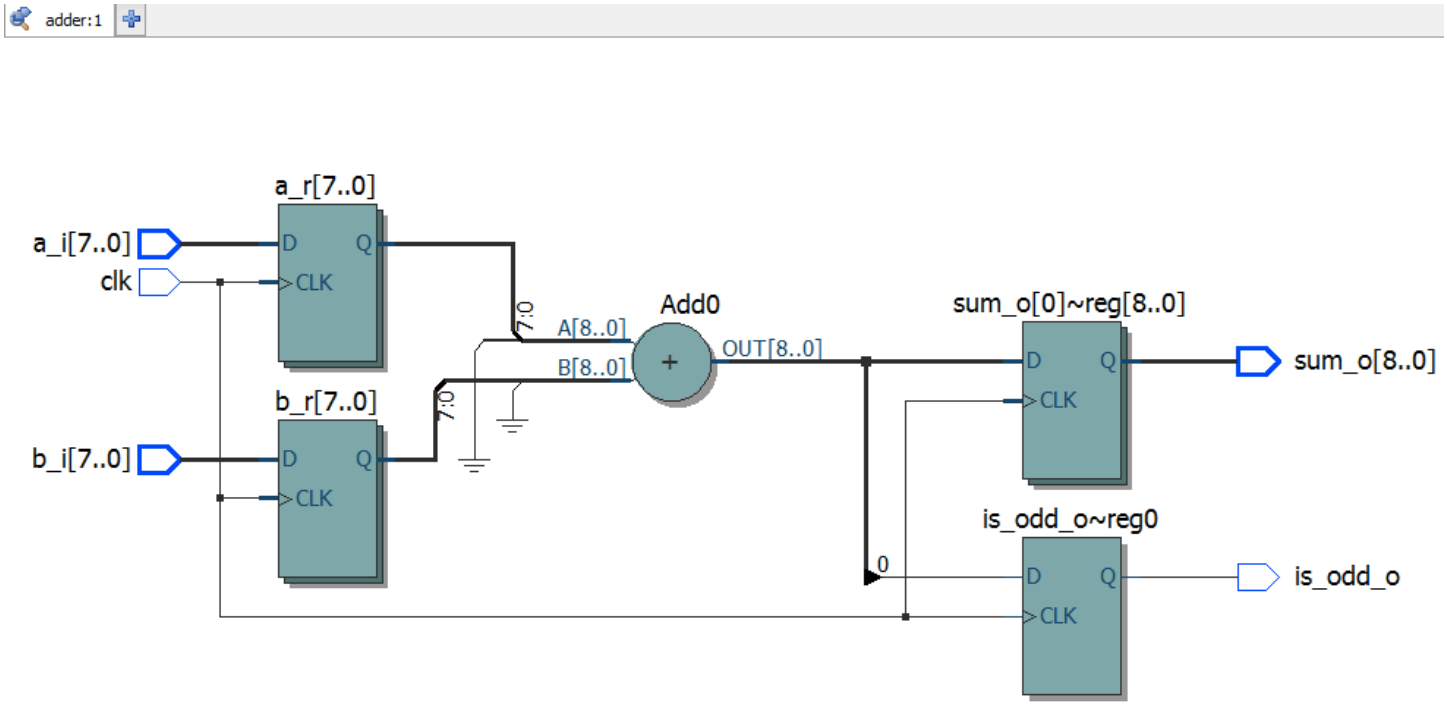
Minh Nguyen
mhn013@ucsd.edu

CSE 141L Lab 1 Report

Q1. Schematic



Q2. Rather than updating is_odd_o from sum_o, I made it so that is_odd_o updates from the least significant bit of sum_next just like sum_o. This way, is_odd_o and sum_o are updating at the same time rather than having is_odd_o update from the last value that sum_o held.



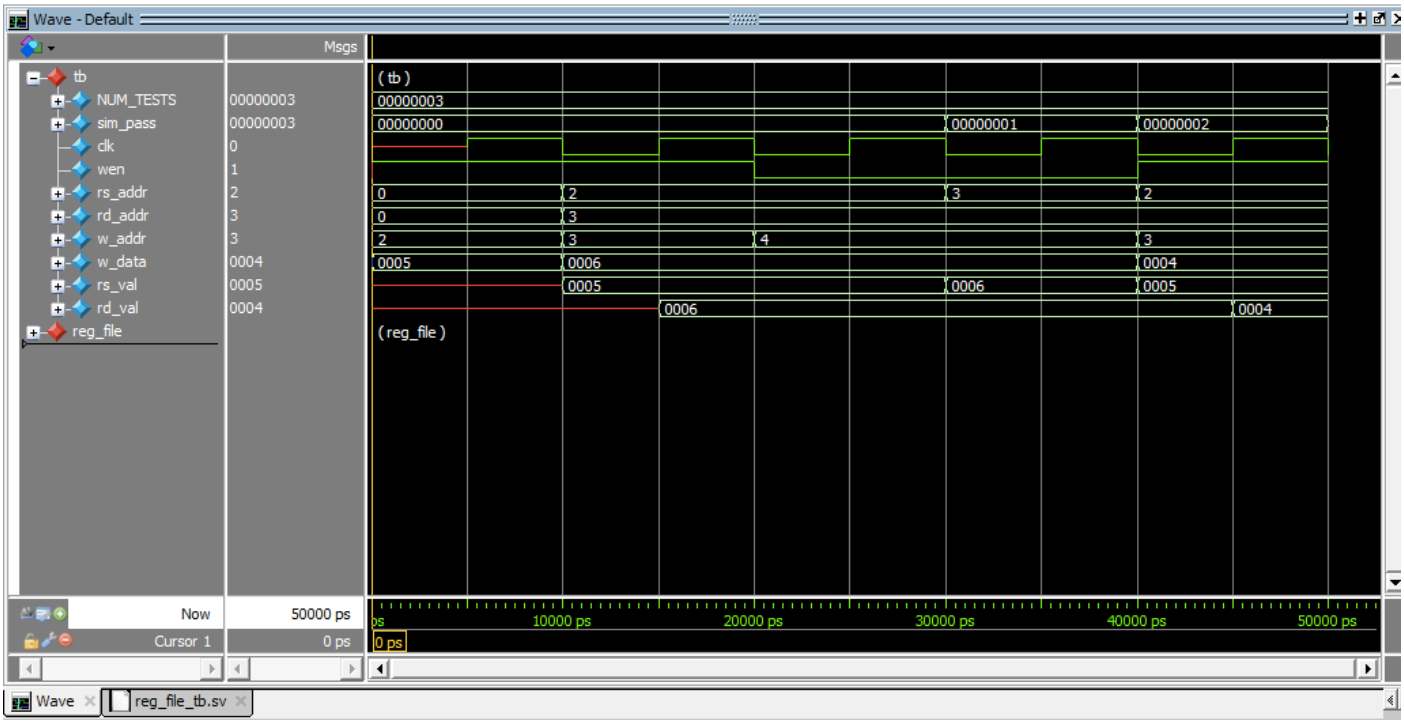
Q3. According to the fitter summary, 26 flip-flops are used for the design of this adder schematic. EP4CE40F29C6 has 39,600 total flip-flops. My design uses 9 combinational functions. EP4CE40F29C6 has 39,600 combinational functions.

Q4. Using the worst case (Slow 1200mV 85C Model), the reported Fmax of my design is 548.85 MHz. Using this Fmax, we obtain the cycle time as follows:
cycle time = 1/(Fmax) = 1/(548.85 MHz) = 1.82 nanoseconds.

Q5. All timings taken from Slow 1200mV 85C Model

Width	Fmax	# of combinational functions	# of registers (flip-flops used)
8 bits	548.85 MHz	9	26
16 bits	416.32 MHz	17	50
32 bits	311.24 MHz	33	98
64 bits	196.16 MHz	65	194

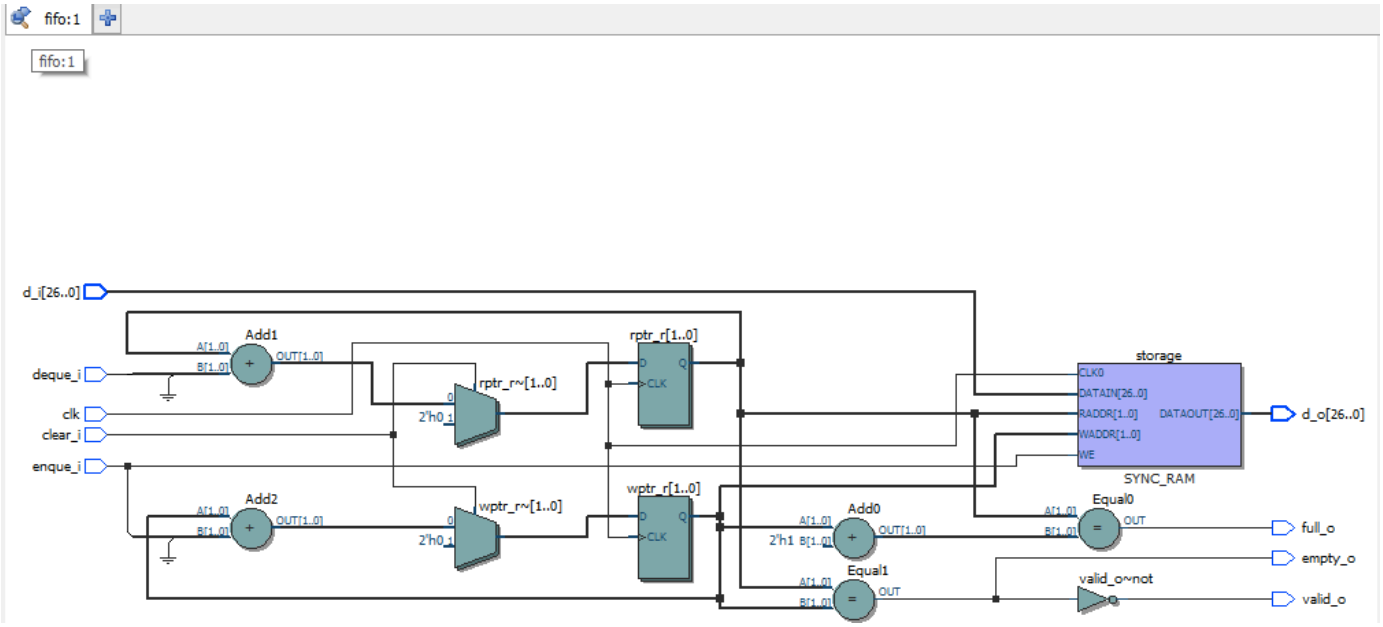
Q6. The first test case is when we are reading from two different registers. This is fairly straightforward as we write the value 5 to register 2 and the value 6 to register 3. Register 2 is being read in rs and register 3 is being read in rd. When we read from two different registers, we should read whatever value is stored in each respective register. This case will pass when the address of rs is 2, the address of rd is 3, the value of rs is 5, and the value of rd is 6.
The second test case is when we are reading the same register on both read ports. In this test case, we are reading register 3 in rs as well as rd. The value read from rs and rd should both display as 6.
Our third test case is when we read to a register while writing to it at the same time. Writes happen only on rising clock edges and so our test case will come on the negative clock edge following the write. This can be seen in the waveform where the number of tests we pass are 3 with the last test case being passed on the final negative clock edge. At this point, the write address and rd address are the same. In addition, the value read in rd is the same as the write data. We can see the value read from rd change because reads are asynchronous.



Q7. All timings taken from Slow 1200mV 85C Model

Configuration	Fmax	# of combinational functions	# of registers (flip – flops) used	# of memory bits
8 16-bit regs	325.63 MHz	168	186	0
16 16-bit regs	1010.1 MHz	0	29	512
32 16-bit regs	985.22 MHz	0	32	1024
32 64-bit regs	973.71 MHz	0	80	4096
256 32-bit regs	876.42 MHz	0	51	4096

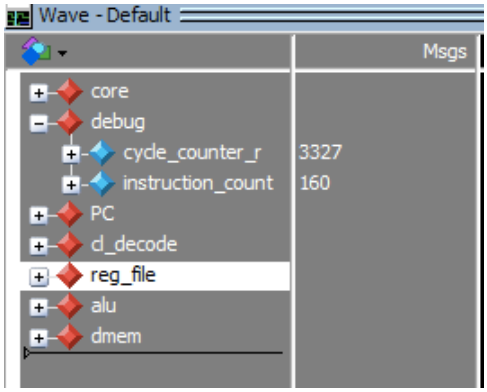
Q8. Schematic, desugared code in fifo.sv



Q9. Added the following line to core/src/tb/core_tb/sim.tcl

add wave -noupdate -group {cl_decode} -radix hexadecimal /core_tb/dut/core1/decode/*

Q10. The number of instructions executed is 160. The cycle counter indicates that the numbers of cycles needed to run the program to completion was 3327.



Q11. For core_flattened, there are 2096 registers, 3918 combinational functions, and 16348 memory bits in the design. The reported Fmax is 54.6 MHz. The cycle time is 1/Fmax which is 1/(54.6 MHz) which is 18.32 nanoseconds. If additional cores require no overhead beyond the resources required by a single core, meaning we didn't have to worry about pins, then the only constraints would be the total logic units and the total memory bits. The total logic elements are at 12% device capacity while the total memory bits are at 1% device capacity. Thus, our max amount of cores we can fit is Math.floor(100/12) = 8. We can instantiate and fit approximately 8 cores on this device. The execution time is obtained by (#cycles * cycle time) which is 3327 cycles * 18.32 ns = 60950.64 ns = 60.96 microseconds.