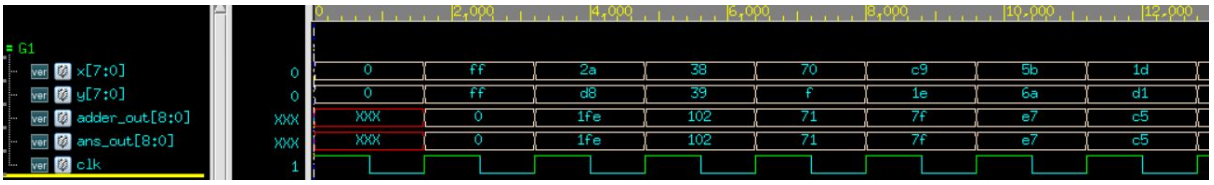
Hw1

B07505026 電機三田昀曜

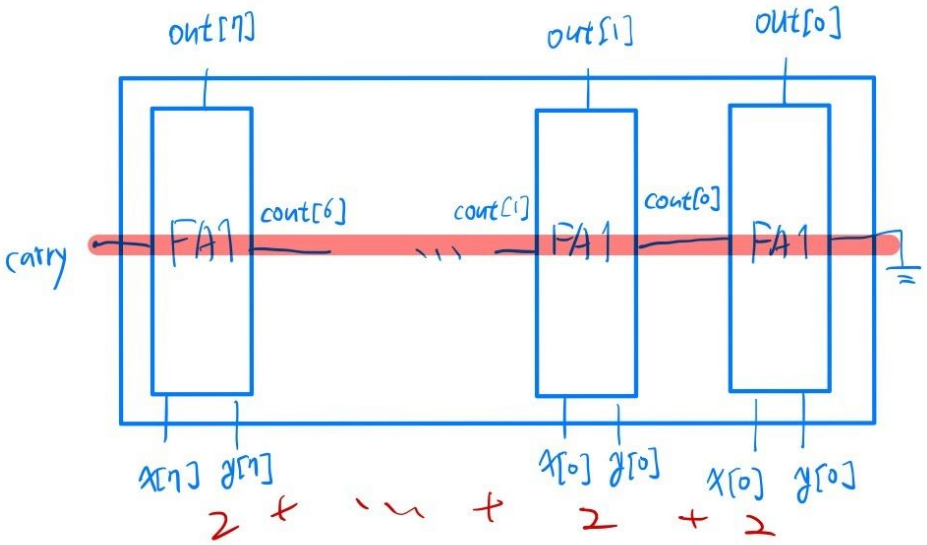
1-CR_Adder

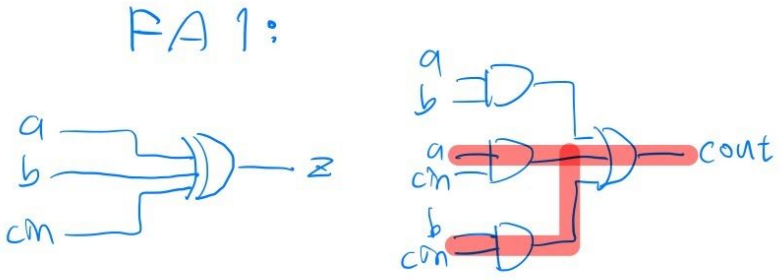
cycle 20.0 waveform:



my design:

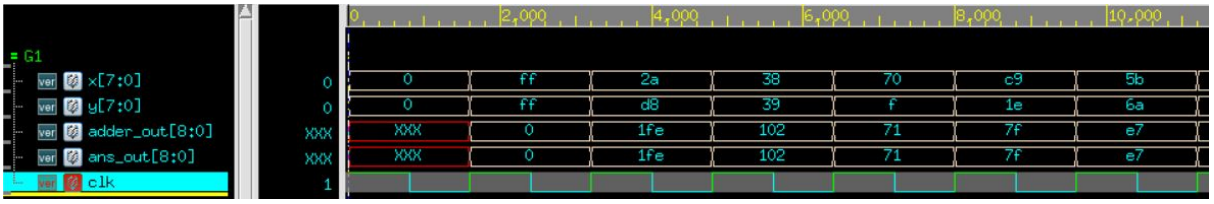
total delay is 2ns*8 = 16 ns (red line is critical path)



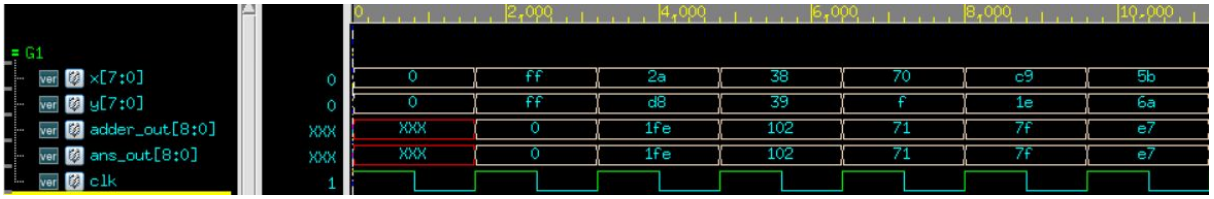


verify:

cycle 16.0 waveform:

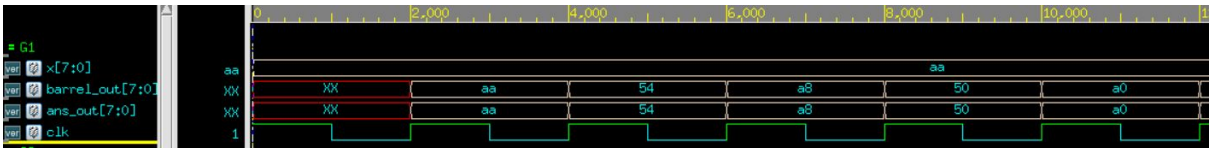


cycle 15.9 waveform:



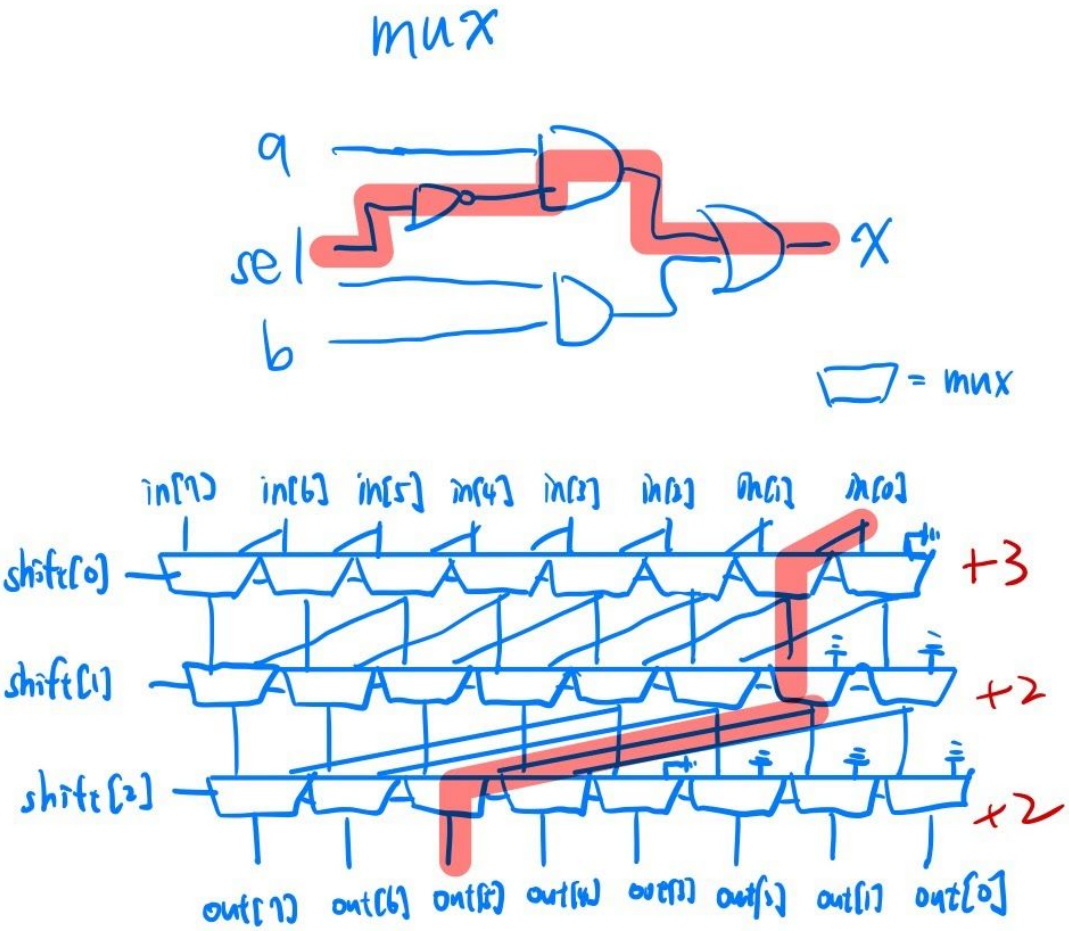
2-barrel_shifter

cycle 20.0 waveform:



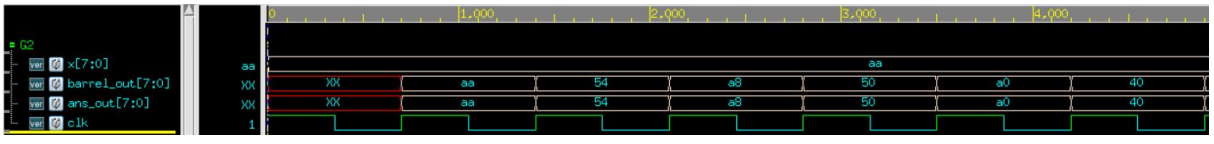
my design:

total delay is 3+2+2 = 7 ns (red line is critical path)

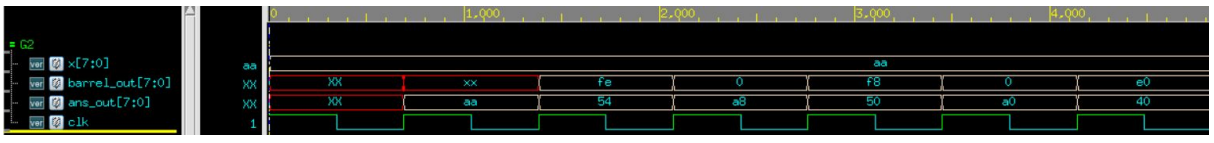


verify:

cycle 7.0 waveform:

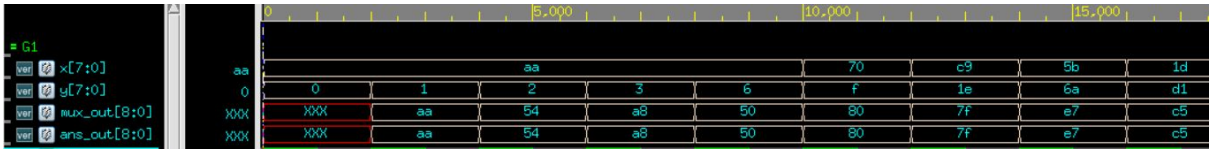


cycle 6.9 waveform:



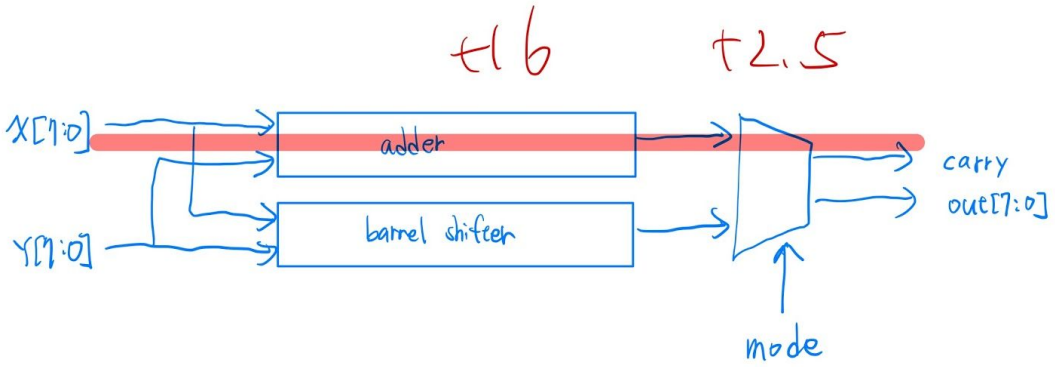
3-ASU

cycle 20.0 waveform:



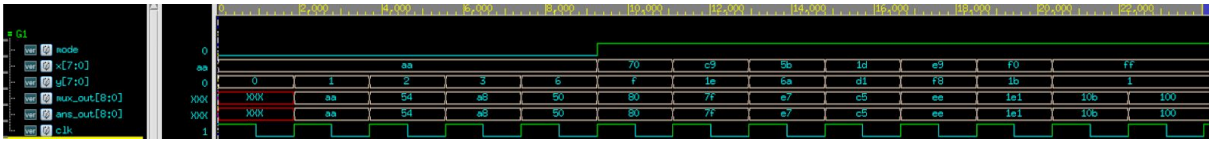
my design:

total delay is 16+2.5 = 16 ns (red line is critical path)

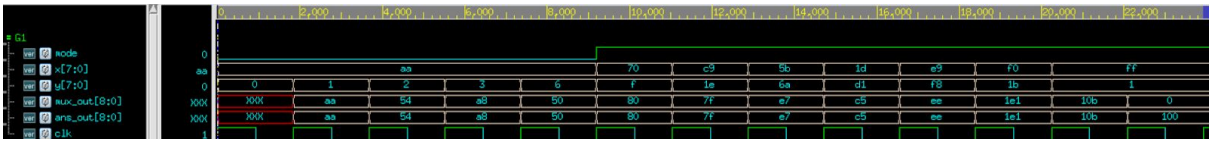


verify:

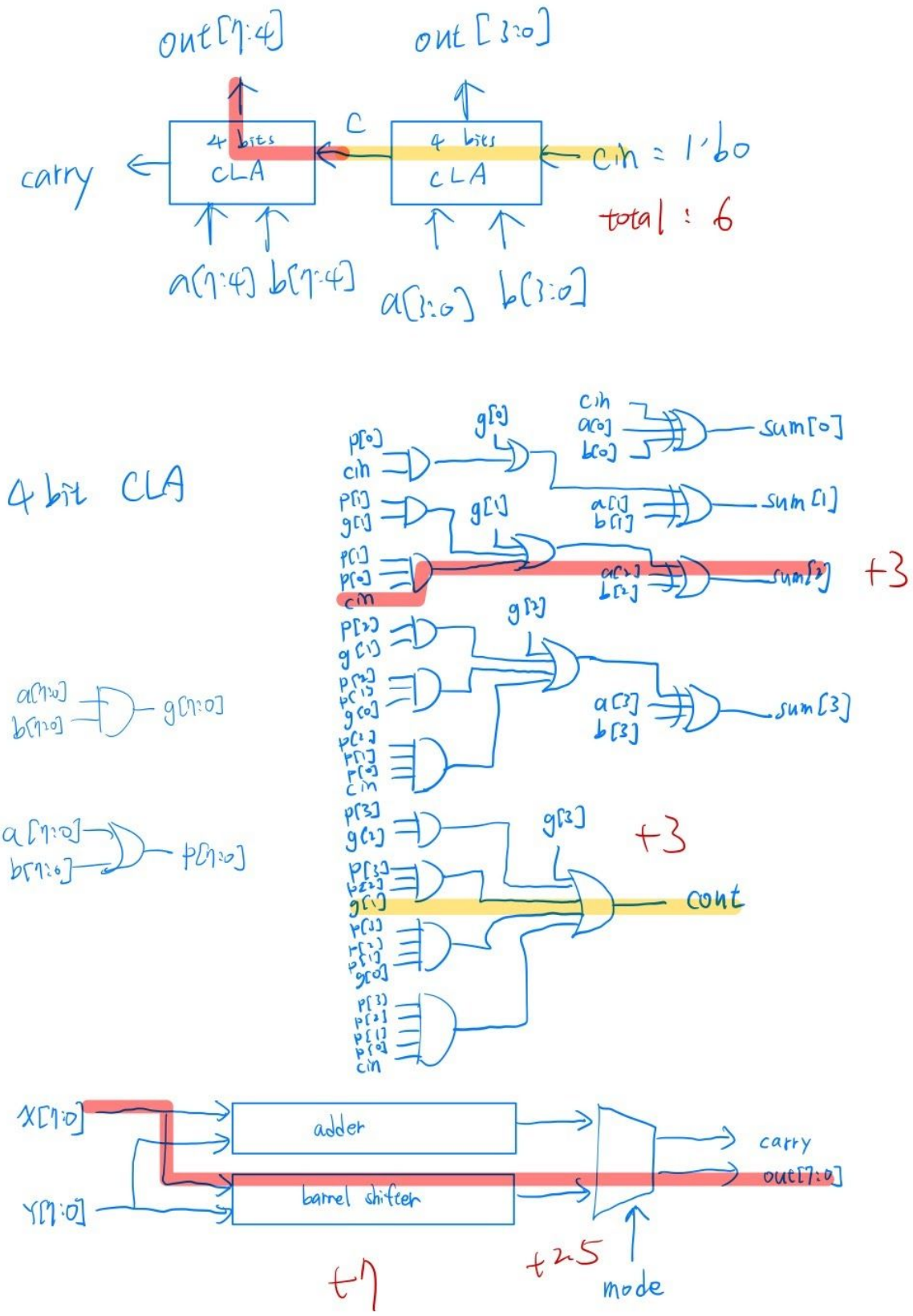
cycle 18.5 waveform:



cycle 18.4 waveform:



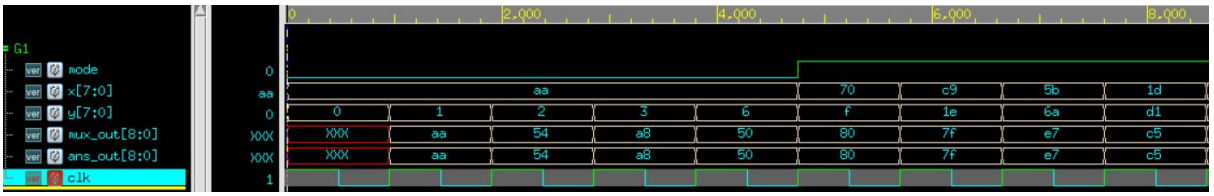
optimize:



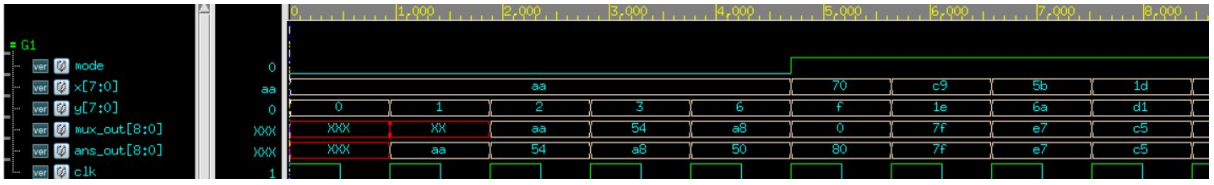
The critical path is due to adder delay. I use 2 4-bits carry-look-ahead to replace the origin one. The total delay of adder_opt

is 6n, which is lower than the total delay of barrel_shifter(7ns). So the critical path of the new design is 9.5ns (7 + 2.5).

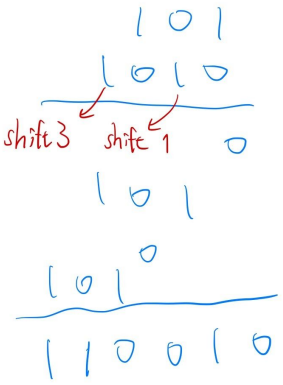
cycle 9.5 waveform:



cycle 9.4 waveform:



multiplication:



Since the multiplication can be viewed as shifting some bits then adding together. And assume we have other registers to store the values we shifted and added. And since the 1, 2, 4 bits barrel_shifter can make 0~7 bits shifting. So we can carefully use mode to control asu to become shifter and adder. That is, we can do unsigned multiplication with the asu.