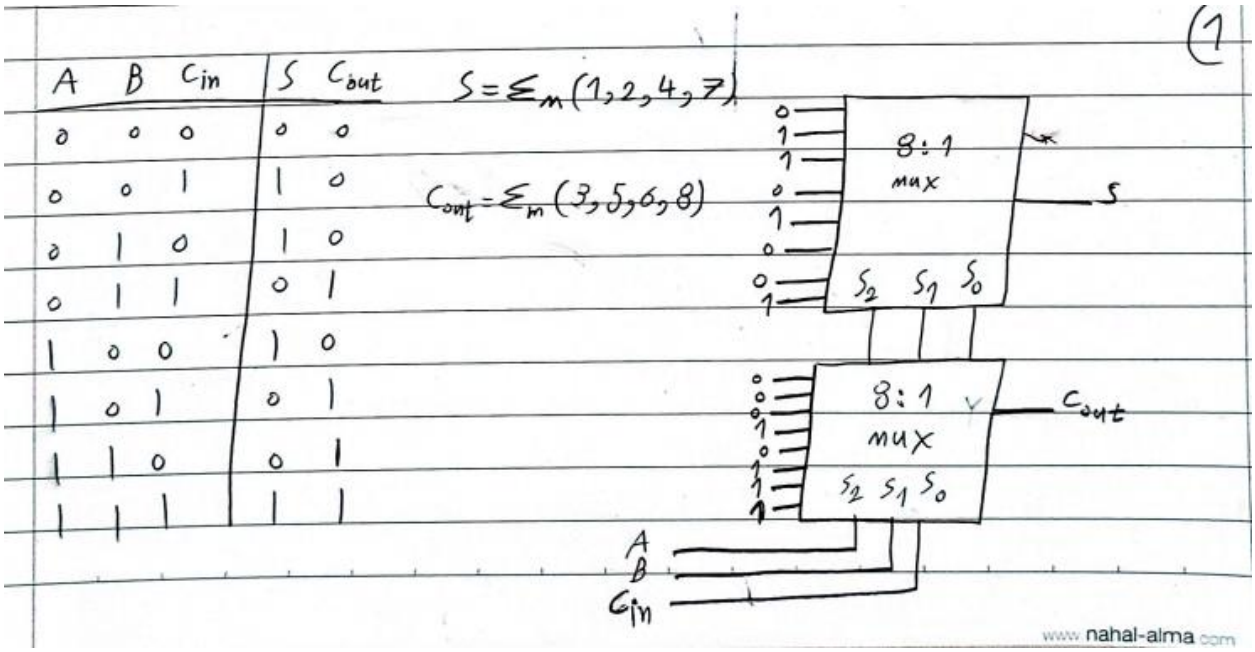
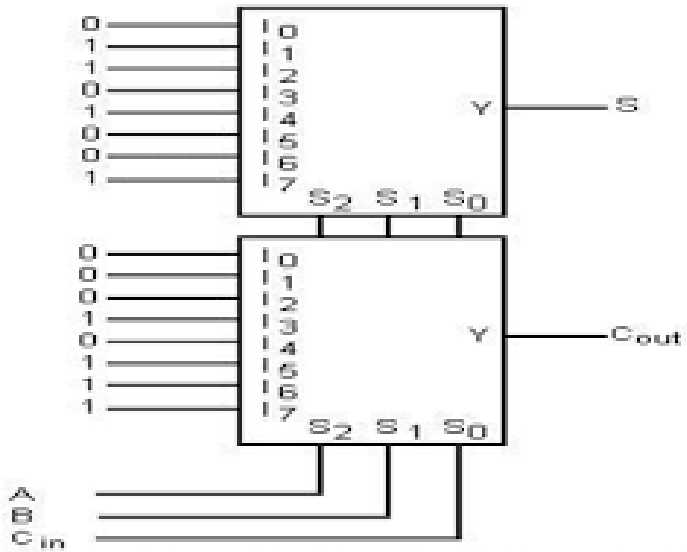
فربد فولادی-98243045

عرفان رفیعی اسکوئی-98243027

سوال اول)

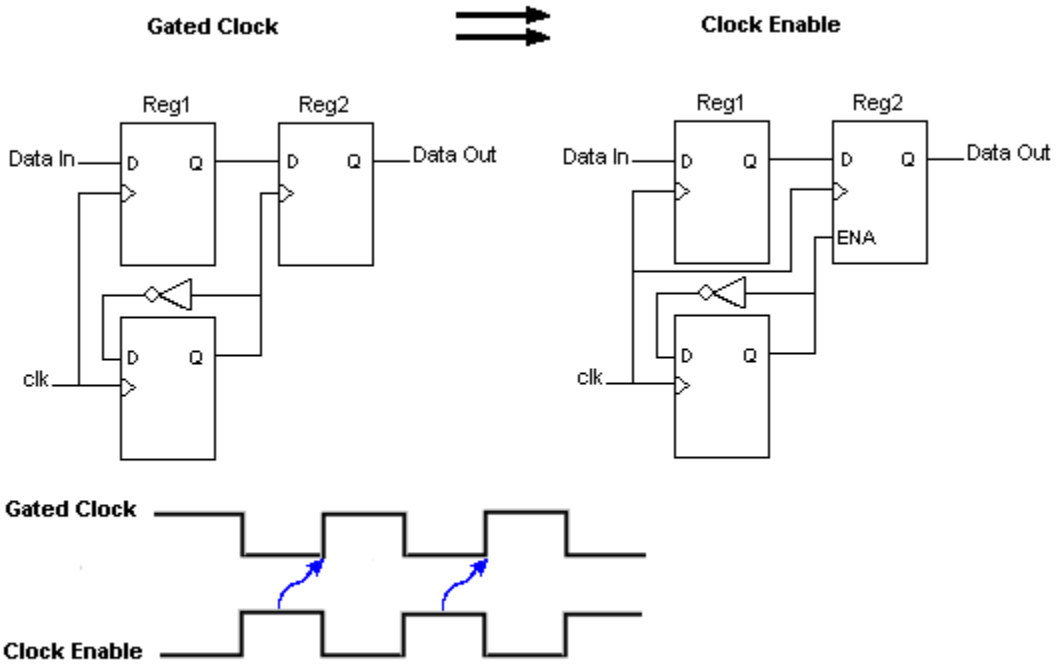




سوال دوم)

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation, by removing the clock signal when the circuit is not in use. Clock gating saves power by pruning the clock tree, at the cost of adding more logic to a circuit. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred.

Although asynchronous circuits by definition do not have a "clock", the term perfect clock gating is used to illustrate how various clock gating techniques are simply approximations of the data-dependent behavior exhibited by asynchronous circuitry. As the granularity on which one gates the clock of a synchronous circuit approaches zero, the power consumption of that circuit approaches that of an asynchronous circuit: the circuit only generates logic transitions when it is actively computing.



سوال سوم)

Sram means the chip programming is held in ram or flops or similar. If you lose power, you lose programming. Usually, the board will have an eeprom to hold the program bitstream and the fpga will load the bitstream automatically at reset.

Fuse fpgas mean the chip is programmed by blowing one time fuses in the fpga. If you cycle power, the chip is still programmed.

If you need fast boot time, you might go with fuse based. Sram fpgas might take 15 seconds or more to load the bitstream. If you dont care about a slow boot time, then go with sram because every time you change the code in a fuse based chip, you need to throw away the old chip.

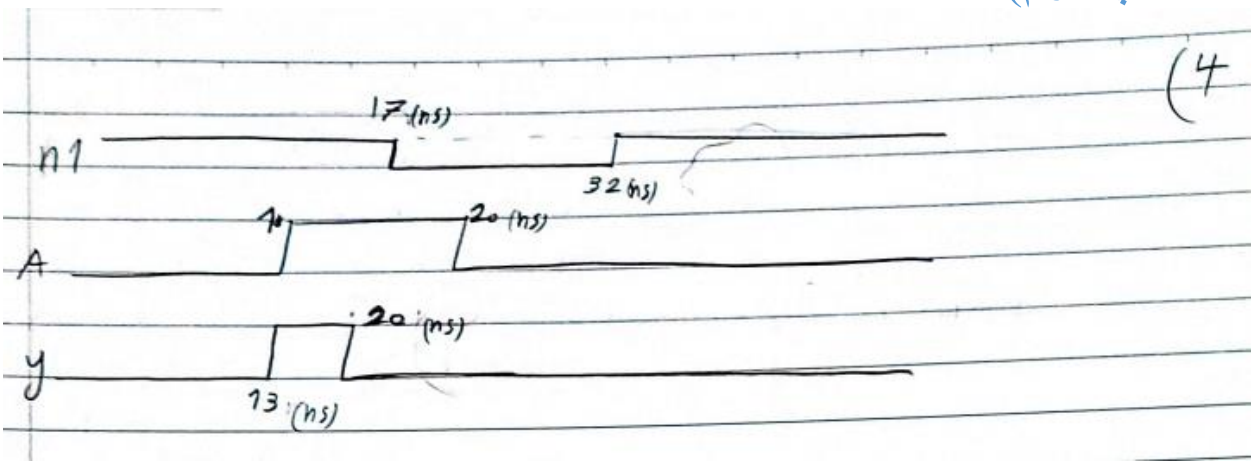
Fuse based might be useful for high reliability applications. They work. They recover from reset immediately, and they are more robust in radiation environments. In high rad areas, radiation can cause a flipflop to invert its stored value. You can correct this with voting logic and such in the design. But if one of the bitstream flip flops inverts, you're in trouble, because the design cant see them.

So, for low cost, low priority stuff, probably go with sram. You can reuse the same chip so its cheaper and easier to develp.

For high reliability or systems that need fast boot/reset times, you probably want fuse based.

SRAM-based FPGAs are volatile and need to be re-programmed at each power-up. This eases prototyping and allows devices to be completely re-configured in-orbit. ... Antifuse-based FPGAs are non-volatile, live at power-up, but one-time programmable, which can present prototyping challenges.

سوال چهارم)



سوال پنجم)