

- 1) You will need to change it to "xor a, #10000000B" as the xor function, put simply, inverts all bits that are xor'ed with a 1 and keeps all bits xor'ed with a 0.

	00101010		10101010
XOR	10000000	XOR	10000000
=	10101010	=	00101010

2)

- a) Harvard.
- b) This architecture has separate memory banks for data and instructions (i.e. separate busses that connect to the CPU for each memory bank).
- c) It is, the core of the system is Harvard based architecture, even if there are some extras added on. This system has the added ability to store instruction memory in data memory banks, the downside is that you cannot execute instructions from this area, only read them. The upside is that this technique could be used to save RAM.

3)

- a) CISK
- b) Generally more clock cycles pass for each ASM mnemonic as you have essentially increased the "sub states" that need to be implemented for the more complex functions.
- c)
  - i) CISK
  - ii) CISK
  - iii) RISK

4) One cycle.

5)

- a) Eight eight-bit registers.
- b) Registers incredibly fast (much faster than RAM), they are SRAM if I am not mistaken, thus using them improves the performance of your program.

6)

- a) Little endian.
- b) My Pc is Windows, so little endian.
- c) When receiving large amounts of information rapidly big endian would be the obvious choice as it can be stored as it is received (assuming it is received in sequential order).

7)

- a) It provides the ADD mnemonic.
- b) It provides the SUB mnemonic.
- c) It provides the MULU mnemonic.
- d) It cannot divide.
- e) You could use the SUB mnemonic repeatedly and subtract the divisor from the specified number (count how many times) and when the number becomes negative (or when the absolute value of the number starts getting bigger again) you know that your counter is the answer to the division problem. (this is assuming that you cannot simply multiply with the divisors inverse)

8) It uses the RL78-S2 core.

9)

- a) Since the register address is between addresses FFE20H and FFF1FH we can just use the short direct addressing, and if you use this you don't need to include the highest four bits.
- b) The only register (essentially) that those operands can dereference to is A so the data must be in A for you to use those operands.