EE309 – Microprocessors

Mid-Semester Exam, Autumn 2013 (IIT Bombay)

Wednesday, September 11, 2013; Duration: 11:00 – 13:00 Hrs; Maximum Marks: 25

1. Can we implement the funtionality of `RET` instruction in 8051 using some other instructions (i.e. without using `RET`). If no, why not? If yes, how (write the set of instructions)? *[2]*

2. Draw the gate/block level schematic to show the hardware for Timer/Counter 0 that can be derived from the TCON and TMOD configurations. *[1]*

3. Write a set of 8051 instructions for initialization of relavant registers/latches/flags/pins etc. to configure the Timer/Counter 0 to generate an interrupt when 1000 negative edges are detected on the T0 (P3.4) pin. You can refer the schematic of problem 2. *[2]*

4. For most 8085 instructions, the number of T-states required for the execution of an instruction is $3n + 1$, where n is the number of times the memory is read from or written to during fetch and execution of the instruction (provided the memory is NOT slow). However, the instruction `PUSH Rp` (a one-byte instruction that copies the contents of the register pair to the top of the stack) requires two T-states more than that suggested by this rule. *[1+1+1]*

   (a) How many T-states does the execution of this instruction require?

   (b) Why does the instruction require more number of cycles?

   (c) Given that the number of T-states required for `POP Rp` (which does the opposite of `PUSH Rp`) is given by the $3n + 1$ rule. Why does the same logic as mentioned for `PUSH Rp` not hold for `POP Rp`.

5. Assume for an 8085 microprocessor, a subroutine call instruction `CALL 4000H` is located in external memory with starting address 10FFH. Give the sequence of byte values available at the A15-A8 pins and the AD7-AD0 pins while this instruction is fetched and executed, if the value of SP just before the execution of this instruction is 1234H. *[4]*

6. You have to interface an 8051 microcontroller with an 8085 microprocessor such that the microcontroller acts like an IO device for the 8085 chip with IO port addresses 00H–7FH and 80H–FFH mapped to the 8051 RAM with addresses 00H–7FH (i.e. MSB of the port addresses can be ignored). For this implementation, the $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$ and READY signals from/to the 8085 chip, and the external interrupt $\overline{INT0}$ (P3.2) pin and some other port/pins of 8051 should be used for handshake etc. You can assume that the port pins of 8051 effectively provide high-impedance if ONEs are written to the corresponding port latches and the clock rates for both 8085 and 8051 are roughly same. *[2+4]*

   (a) Draw neatly the schematics showing a possible way to interface the two (you can assume that the external memory device used by 8085 is not slow and there is no need to show it in your schematics). You can also use some additional logic gates if required.

   (b) Write a set of initialization instructions and the $\overline{INT0}$ interrupt service subroutine for 8051 to implement the above functionality. The interrupt should be used in the edge triggered mode.

7. We wish to implement a counter in 8051 that counts from 0–9999 in decimal format. The upper two decimal digits are stored in the two nibbles of R1 register and the lower two decimal digits are stored in the two nibbles of R0 register (assume that flags RS1=0, RS0=0). Write a 8051 subroutine that increments this counter by one every time it is called. If the previous value was 9999, the counter should roll over to 0 and OV flag should be raised to 1. Also, assume that both R0 and R1 contain 0 initially. [Hint: Use DA instruction] *[3]*

8. Write a set of 8051 instructions to subtract the value 1234H from the contents of DPTR (the final result should be saved in DPTR only). If a borrow is required, the carry flag should be raised to 1 by the program. *[2]*

9. Assume that the P1.0 pin of an 8051 chip has been connected to ground externally using a wire. Consider the following instructions: *[1+1]*

```
SETB P1.0
CPL P1.0
MOV C, P1.0
```

(a) What will be the state of the Carry flag after the above 3 instructions are executed and why?

(b) What will be the state of Carry flag in case the wire to ground is removed after execution of the second instruction itself and why?

```
MOV  A, #1234H
MOV  B, A
MOV  DPTR, #data 16
MOV  A, @DPTR
SUBB A, B
MOV  @DPTR, A
```