

Jonathan Valvano

February 16, 2004, 1:00pm-1:05pm

Question 1. Write *friendly* C code that makes PT7 an output. The answer to this problem does not need to be a complete function, just a C code fragment.

```
DDRT = DDRT | 0x80;
```

or

```
DDRT |= 0x80;
```

Question 2. Write *friendly* C code that sets PT7 low. The answer to this problem does not need to be complete function, just a C code fragment.

```
PORTT = PORTT & 0x7F; // 9S12C32 substitute PTT for PORTT
```

or

```
PORTT = PORTT & ~0x80;
```

or

```
PORTT &= 0x7F;
```

or

```
PORTT &= ~0x80;
```

Question 3. Which three events will trigger a Real Time Interrupt (RTI)? I.e., which events if true will cause a RTI interrupt to happen? *Give three letters where the order of events does not matter.*

- A) Software sets **RTIE** to 1, **arm**
- F) Software sets the **I** bit to 0, **enable**
- D) Timer hardware sets **RTIF** to 1, **flag**

Why these are not true:

- B) Software sets **RTIE** to 0, **disarm**
- C) Software sets **RTIF** to 1, software can not set flag
- D) Software sets **RTIF** to 0, acknowledge, software does this in ISR
- E) Software sets the **I** bit to 1, **disable**
- G) Timer hardware sets **RTIE** to 1, hardware can not change this bit
- H) Timer hardware sets **RTIE** to 0, hardware can not change this bit
- J) Timer hardware sets **RTIF** to 0, hardware can only set the flag

Question 4. Which three events occur automatically in hardware as an interrupt is processed, making the thread switch from foreground to background? *This order matters.*

- D) The registers are pushed on the stack, notice it pushes CCR with I=0
- A) The **I** bit is set to 1, automatically disables before running ISR
- F) The interrupt vector is loaded into the PC register, vector contains the address of ISR

Why these are not true:

- G) The **RTIE** bit is cleared to 0, the arm bit is under software control
- B) The **I** bit is cleared to 0, this will be cleared at the end of the ISR by the **rti** instruction
- C) The **RTIF** bit is cleared to 0, acknowledge occurs as a software action in the ISR
- E) The interrupt vector is loaded into the SP register, SP is the stack pointer