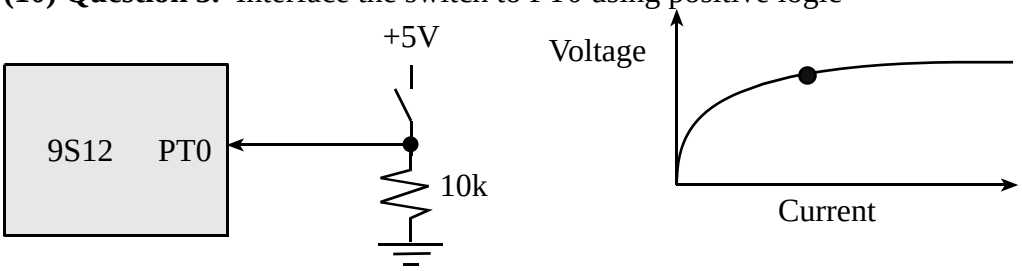
(5) Question 1. $1210 = 1*16^3+2*16^2+1*16^1 = 1*4096+2*256+1*16 = 4096+512+16 = 4624$

(6) Question 2.
Part a) Data flow graph
Part b) Basis
Part c) Memory mapped I/O

(6) Question 3. Consider `ldaa #-100 adda #90`
To calculate the overflow (V) bit, convert to signed, and operate. -100+90 = -10 is correct, so V = 0
To calculate the carry (C) bit, convert to unsigned, and operate. 156+90 = 246 is correct, so C = 0

(5) Question 4. 2^{10} is about 10^3, so 2^{30} is about 10^9, which is 9 decimal digits.

(10) Question 5. Interface the switch to PT0 using positive logic

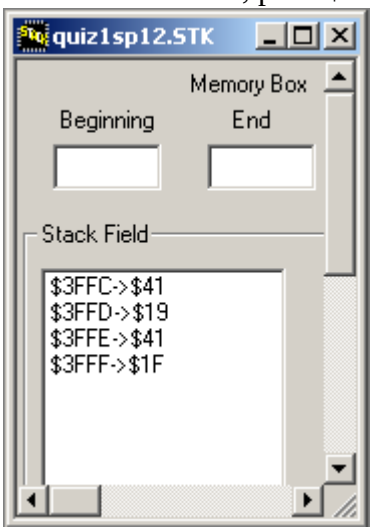


(5) Question 6. Current is exponentially related to voltage

(5) Question 7. `$5000 D001 subb $0001`

R/W	Addr	Data	Changes to A,B,X,Y,S,PC,IR,EAR
R	$5000	$D0	IR=$D0, PC=$5001
R	$5001	$01	EAR=$0001, PC=$5002
R	$0001	$12	B = $34-$12 = $22

(4) Question 8. Subroutine return address is the address of the instruction after the **`bsr`**. Push $411F when **`Test`** called, push $4119 when **`Delay`** called. 16-bit numbers are stored big endian.



(5) Question 9. C code to create a variable named **`Position`** with range -128 to +127

```
char Position;
```

or

```
signed char Position;
```

(10) Question 10. You only have to push one of them

```
  pshd
  tfr Y,D   ; Y goes to D
  tfr X,Y   ; X goes to Y
  pulx      ; D goes to X
```

It works, but is less efficient to push all three

```
  pshd   ; D on stack
  pshx   ; X,D (X on top)
  pshy   ; Y,X,D (Y on top)
  puld   ; Y goes to D
  puly   ; X goes to Y
  pulx   ; D goes to X
```

(20) Question 11. Two positive logic switches are connected to PT2 and PT0, and one positive logic LED is connected to PT5. Turns on the LED if exactly one of the two switches is on.

```
      org  $4000
main  lds  #$4000    ;optional here because stack not used
      bset DDRT,#$20 ;PT5 output
      bclr DDRT,#$05 ;PT2 and PT0 inputs
loop  ldaa PTT
      anda #$05      ;RegA is 00,01,04 or 05
      cmpa #$04      ;PT2 set, PT0 clear
      beq  On
      cmpa #$01      ;PT0 set, PT2 clear
      beq  On
Off   bclr PTT,#$20  ;LED off
      bra  loop
On    bset PTT,#$20  ;LED on
      bra  loop
      org  $FFFE
      fdb  main      ;reset vector
```

A second possible solution without conditionals, PT5 = PT2^PT0

```
      org  $4000
main  lds  #$4000 ;optional here because stack not used
      ldaa DDRT
      oraa #$20   ;PT5 output
      anda #$FA   ;PT2 and PT0 inputs
      staa DDRT
loop  ldaa PTT    ;read Bit 0
      lsla
      lsla        ;bit2 has PT0 value
```

```
      eora PTT    ;bit2 has PT2^PT0
      lsla
      lsla
      lsla        ;bit5 has PT2^PT0
      anda #$20   ;RegA only has PT2^PT0 in bit 5
      ldab PTT
      andb #$DF   ;RegB has original PT7,PT6,PT4-PT0
      aba         ;combine
      staa PTT    ;PT5 = PT2^PT0
      bra  loop
      org  $FFFE
      fdb  main   ;reset vector
```

A third possible solution with fewest instructions

```
      org   $4000
main  lds   #$4000       ;optional here because stack not used
      bset  DDRT,#$20    ;PT5 output
      bclr  DDRT,#$05    ;PT2 and PT0 inputs
loop  brset PTT,#$04,Is2 ;go to is2 if PT2 is high
no2   brset PTT,#$01,On  ;turn on if PT2=low and PT0=high
Off   bclr  PTT,#$20     ;LED off
      bra   loop
is2   brset PTT,#$01,Off ;turn off if PT2=high and PT0=high
On    bset  PTT,#$20     ;LED on
      bra   loop
      org   $FFFE
      fdb   main         ;reset vector
```

(20) Question 12. Write a C program that controls a kidney dialysis pump.

```
void main(void){
  DDRP = 0xFF;  // output power to pump
  DDRT = 0x00;  // input flow rate in ml/min
  while(1){
    if(PTT < 150){ // too slow
      if(PTP < 255){
         PTP++;    // increase power
      }
    }
    if(PTT > 150){ // too fast
      if(PTP > 0){
         PTP--;    // decrease power
      }
    }
  }
}
```

*Just for fun, open assemble and run the **motor.uc** example in TExaS.*