

1- In Tm4c123 Tiva C board, switches are connected to ____, ____, and are ____ logic respectively

- a- PF1, PF4, Positive b- PF2, PF3, Negative c- PF0, PF4, Negative

2- In a microprocessor, the register which holds the address of the next instruction to be fetched is ____.

- a- R12 b- R15 c- R14

3- the stack is used for ____.

A: Storing the program return address whenever a sub-routine jump instruction is executed.

B: Transmitting and receiving input-output data.

C: Storing all important CPU register contents whenever an interrupt is to be serviced.

D: Storing program instructions for interrupt service routines.

4- most of Tm4c123 register are ____ width

- a- 32-bit b- 16-bit c- 8-bit

5- PSR register can monitor ____

- a- zero status b- overflow status c- carry status d- all of previous

6- word-aligned mean that ____ must be zero .

- a- Most significant bit b- two least significant bit c- least significant bit

7- UART in Tm4c123 is ____

- a- Half-Duplex b- Full-Duplex c- Simplex

8- Usually frame of UART used is

- a- start bit, data bits, parity bit
b- start bit, data bits, 2 stop bits
c- start bit, data bits, 1 stop bit

9- UART is ____

- a- Universal Analog Receiver Transmitter b- Universal Synchronous Receiver-Transmitter
c- Parallel Communication d- None of previous

10- which of following shows that system clock has been enabled for Port F

- a- GPIO_PORTF_DIR_R = 0x01
b- GPIO_PORTF_DEN_R = 0x01
c- SYSCTL_RCGCGPIO_R = 0x20
d- SYSCTL_PRGPIO_R = 0x20

11- the following part of Embedded C code has an error in its sequence of execution, what is the right sequence (neglect clock enable status checking)

```
1 GPIO_PORTF_DEN_R |= 0x12;           // Enable PF1 and PF4 as a digital GPIO pins
2 GPIO_PORTF_CR_R = 0x01;             // Enable GPIOPUR register enable to commit
3 SYSCTL_RCGCGPIO_R |= 0x20;          /* enable clock to GPIOF */
4 GPIO_PORTF_LOCK_R = 0x4C4F434B;      // unlock GPIOCR register
5 GPIO_PORTF_PUR_R |= 0x10;            // Enable Pull Up resistor PF4
6 GPIO_PORTF_DIR_R |= 0x02;            // set PF1 as an output and PF4 as an input pin
```

- a- 3,4,1,5,2,6 b- 3,4,2,5,6,1 c- 5,4,1,6,2,3 d- 2,3,4,1,5,6

12- Which of the following is the correct instruction for toggling bit 3 of port A data register?

- a- GPIO_PORTA_DATA_R |= 0x08
- b- GPIO_PORTA_DATA_R &= ~0x08
- c- GPIO_PORTA_DATA_R ^= 0x08

13- which statement is used to clear pins 1,2,3 of port F without affecting rest of pins

- a- GPIO_PORTA_DATA_R &= ~0x0E
- b- GPIO_PORTA_DATA_R = ~0x0E
- c- GPIO_PORTA_DATA_R ^= ~0x0E
- d- GPIO_PORTA_DATA_R &= 0x0E

14- assuming Baud rate =115200 , what is the value of UARTIBRD and UARTFBRD respectively (using SysClk =16Mhz)

- a-104,11 b-11,104 c-44,8 d-8,44

15- where are local variables stored in memory ?

- a- ROM b-RAM c- Stack d- none of previous

16- Assuming 9600 Baud Rate , what's time required to transmit "System Error" via UART using 1stop-bit

- a-10ms b-15ms c-12.5ms d-11.25ms

17- for previous problem , if Baud Rate modified to be 115200 , time required will decrease by ____

- a- 8.6% b- 57% c-17.5% d- 91.6%

18- which of the following can be used as UART Rx &Tx

- a- PA3,PA4 b-PA0,PA1 c-PD2,PD3 D-PE2,PE3

19-assuming an integer variable "Z"=(53)₁₀ , how to clear bit 4 ,what the new value in Hex ?

- a- z&=(1<<4)(16)₁₆
- b- z|=~(1<<4) (12)₁₆
- c- z&=~(1>>4)(37)₁₆
- d-z&=~(1<<4)..... (25)₁₆

For the following program , when you run it there is an error

20- the error is

- a-clock is enabled for different port
- b- PA1,PA2,PA3 are not set as output
- c- port F is locked
- d- port F is Unlocked

21- suppose that error has been fixed , what the out put if Switch 2 pressed ..

- a-LED is Green
- b-LED is Red
- c-Led is Blue
- d- LED is OFF

22- if GPIO_PORTF_DIR_R modified to be 0x11 , then we pressed Switch 2 again

- a-LED is Green
- b-LED is Red
- c-Led is Blue
- d- LED is OFF

```
1  int main()
2  {
3      SYSCTL_RCGCGPIO_R=0x02 ;
4      while ((SYSCTL_PRGPIO_R&0x02)==0){} ;
5      GPIO_PORTF_CR_R |= 0x0E;
6      GPIO_PORTF_DIR_R |= 0x0E;
7      GPIO_PORTF_DEN_R |= 0x0E;
8      while(1)
9      {
10         SW1 = GPIO_PORTF_DATA_R&0x10;
11         SW2 = GPIO_PORTF_DATA_R&0x01;
12         if (!SW1 && !SW2) {
13             GPIO_PORTF_DATA_R = 0x04;}
14         else if (!SW1) {
15             GPIO_PORTF_DATA_R = 0x02;}
16         else if (!SW2) {
17             GPIO_PORTF_DATA_R = 0x08;}
18         else {
19             GPIO_PORTF_DATA_R = 0x00;}
20     }
21 }
```

- a- The receiver could appear to get TWO input frames for every one frame transmitted.
- b- The receiver could appear to get ONE input frames for every one frame transmitted.
- c- The receiver could appear to get 4 input frames for every one frame transmitted.
- d- the receiver will not accept the data

a- 0x00 b-0x09 c-0x90 d-0xFF

a-0x0000 b-0x0001 c-0x0030 d-0x0300

a- UARTRSR = 0x01 b- UARTRSR = 0x02 c- UARTRSR = 0x08

a- 16 , 12 b- 12 , 8 c-8 , 12 d-12 , 10

a- 1,2, b-9,10 c-6,7

a- 0x010 b-0x000 c-0x301

a- True b- False

a- 0x010 b-0x000 c-0x301

```

1  int UART_Init()
2  {
3      SYSCTL_RCGCGPIO_R |= 0x001 ;
4      SYSCTL_RCGCUART_R |= 0x001 ;
5      UART0_CTL_R = 0x000 ;
6      UART0_IBRD_R = 104 ;
7      UART0_FBRD_R = 11 ;
8      UART0_CC_R = 0;
9      UART0_CTL_R = 0x0300;
10     UART0_CRH_R = 0x000 ;
11     GPIO_PORTA_AMSEL_R &= ~0x003;
12     GPIO_PORTA_AFSEL_R |= 0x003;
13     GPIO_PORTA_DEN_R |= 0x003;
14 }

```

- a- Check if the buffer is full
- b- Check if the buffer is not empty
- c- Check if the buffer is empty
- d- None of the other answers

- a- DATA_R is Empty
- b- DATA_R is Full
- c- DATA_R has new data received
- d- b&c

a-5s b-10ms c-7.5ms d-5ms

a-40 b- 4 c- 4000 d-3

Answers

1. C
2. B
3. A
4. A
5. D
6. B
7. B
8. C
9. D
10. D
11. B
12. C
13. A
14. D
15. C
16. C
17. D
18. B
19. D
20. C
21. A
22. D
23. A
24. C
25. D
26. C
27. B
28. B
29. C
30. A
31. A
32. A
33. D
34. D
35. C