Embedded Systems Midterm

Study online at https://quizlet.com/_fojse8

1.	**True**	MCU (Microcontroller unit) consists of a CPU (Central Processing Unit), memory and some other peripherals.
2.	**simple peripherals such as buttons and LEDs**	Which one of the following is not part of MSP432 MCU? - simple peripherals such as buttons and LEDs - Blocks that are in charge of reset, clock and power - Several timers - Different types of memory including ROM, SRAM and flash -Analog interface including ADC (Analog to Digital Converter) - Digital interface including UART SPI, I2C, I/O ports - CPU
3.	**- is the emulator on the launchpad board that helps program/debug the main microcontroller - the main microcontroller on the Launchpad board**	XDS110 is the is the emulator on the launchpad board that helps program/debug the main microcontroller whereas MSP432 is the main microcontroller on the Launchpad board .
4.	**- MSP432 is on the Launchpad.**	Which statement is true? - MSP432 is on the Launchpad. - Launchpad is on the MSP432. - XDS110 is on the MSP432. - MSP432 is on the XDS110.
5.	**Boosterpack**	On what component is the tri-color LED that can be driven by PWM - Boosterpack -LaunchPad - MSP432 chip
6.	**Launchpad**	On what component is the tri-color LED cannot be driven by PWM

		- Boosterpack -LaunchPad - MSP432 chip
7.	**Boosterpack**	On what component is Joystick - Boosterpack -LaunchPad - MSP432 chip
8.	**MSP432 Chip**	On what component is the ADC (Ana-log-toDigital Converter) - Boosterpack -LaunchPad - MSP432 chip
9.	**Boosterpack**	On what component is the large S1 and S2 buttons found - Boosterpack -LaunchPad - MSP432 chip
10.	**MSP432 chip**	On what component is the Timer_A found - Boosterpack -LaunchPad - MSP432 chip
11.	**MSP432 chip**	On what component is the UART found - Boosterpack -LaunchPad - MSP432 chip
12.	**MSP432 chip**	On what component is Timer32 found - Boosterpack -LaunchPad - MSP432 chip
13.	**Launchpad**	On what component is the micro-USB port found

		- Boosterpack -LaunchPad - MSP432 chip
14.	**Launchpad**	On what component is the Debugger chip found - Boosterpack -LaunchPad - MSP432 chip
15.	**little-endian**	If the first address the integer resides is A, it is going to take up A, A+1, A+2, and A+3 in the memory. If the least significant byte is stored at address A, followed by the next least significant byte at A+1 and so on, it is called
16.	**big-endian**	if the most significant byte is at address A, followed by the next significant byte at address A+1 and so on, it is called
17.	**True**	Choose True or False: 1) A struct in the C programming language is a composite data type (or record) declaration that defines a physically grouped list of variables to be placed under one name in a block of memory, allowing the different variables to be accessed via a single pointer.
18.	**True**	Choose True or False: In C, a struct can be returned value of a function.
19.	**True**	Choose True or False: In C, it is possible to pass a struct to a function by its pointer and get it modified by the function.

20.	**True**	Consider the below code snippet, choose the true and false statements. typedef struct {int x1, y1, x2, y2; //(x1,y1) is the top left and (x2, y2) is the right bottom corners of the rectangle } rectangle_t; It is possible to declare a rectangle as: rectangle_t myRec = {1, 1, 3, 4};
21.	**False**	In the below snippet, myRec_p is the pointer pointing to myRec: rectangle_t myRec = {1, 1, 3, 4}; rectangle_t *myRec_p = *myRec;

22.	**True**	The below snippet does NOT have any syntax error:
		rectangle_t someRec;
		rectangle_t *myRec_p = &someRec;
		myRec_p->x1 = 5;

23.	**False**	The below snippet of code does NOT have any syntax error: rectangle_t myRec; myRec->x1 = 5;
24.	**True**	It is OK to declare a function as below (The function gets 4 integers and builds a rectangle type and returns it as the output of the function): rectangle_t GetRectangle(int x1, int x2, int y1, int y2)
25.	**It can be either -52 or 204.**	Consider an 8-bit number A is equal to 11001100. What can we say about value of A in decimal?
26.	**-2^15**	Which one of the following numbers can fit in a 2's complement 16-bit reg-

		ister? - -2^15 - 2^15 - -2^16 - (2^16)-1
27.	**8**	How many bits in a signed character?
28.	**32**	How many bits in an integer?
29.	**We do not have sufficient information to answer this question with yes or no.**	Does adding 0x0fffefee and 0x0effffff result in an overflow?
30.	**-2^15**	What is the lowest number that can be stored in a signed short? (Hint: A C short is two bytes.)
31.	**2^32-1**	What is the highest number that can be stored in an unsigned long? (Hint: A C long is four bytes.)
32.	**2^4**	Which one is bit 4 mask or BIT4?
33.	**(1<<23)**	Which one is bit 23 mask or BIT23?
34.	**True**	True or False: A port is a combination of pins.
35.	**False**	True or False: MSP432 has 1 port, but many pins.
36.	**False**	True or False: A port is used as output and a pin as input.
37.	**False**	True or False: Port and pin are two names of the same thing.

38.	**one**	In an MSP432 digital I/O module, there are how many Schmitt triggers?
39.	**two**	In an MSP432 digital I/O module, there are how many 2-input logic gates?
40.	**three**	In an MSP432 digital I/O module, there are how many multiplexers?
41.	**two**	In an MSP432 digital I/O module, there are how many switches
42.	**True**	True or False: Two gates cannot share a certain wire as their output as it may cause a short in the circuit.
43.	**((a>>n) == 0)**	Which expression could be untrue if the nth bit of the integer a is 0? - ((a & (1<<n)) == 0) - !(a&(1<<n)) -((a>>n) == 0) - ~(a&(1<<n))
44.	**(a << 4) + (a << 1) + a;**	A C expression to compute a * 19 (a times 19) is: (Recall that one time shift to left doubles an integer.)
45.	**multiplexer**	What selects a certain input among multiple inputs
46.	**tri-state buffer**	What is able to cut off a path in the circuit
47.	**schmitt trigger**	What can act as a basic analog to digital converter
48.	**False**	

		True or False: In MSP432, a digital I/O cannot be used for any other purpose such as an analog input.
49.	**True**	In MSP432, any digital I/O pin can be configured to be a digital input.
50.	**True**	In MSP432, any digital I/O pin can be configured to be a digital output.
51.	**True**	In MSP432, a digital input can use any of these three options: 1) attached pull-up resistor, 2) attached pull-down resistor, 3) no attached resistor.
52.	**False**	True or false: A programmer is not able to change the functionality of a digital I/O after the chip is manufactured with a certain configuration.
53.	**False**	True or False: A programmer is not able to change the functionality of a digital I/O after the chip is placed on a PCB board with certain peripherals and a specific configuration.
54.	**True**	True or False: A programmer is able to change the configuration of a digital I/O at any time.
55.	**0V**	Consider a Schmitt trigger with two thresholds at 1.4V and 1.6V. Also, assume the logic 0 is 0V, and logic 1 is 3V. What is the output when the input is 0.2V?

56.	**We cannot tell. It depends**	Consider a Schmitt trigger with two thresholds at 1.4V and 1.6V. Also, assume the logic 0 is 0V, and logic 1 is 3V. What is the output when the input is 1.55V?
57.	**3V**	Consider a Schmitt trigger with two thresholds at 1.4V and 1.6V. Also, assume the logic 0 is 0V, and logic 1 is 3V. What is the output when the input is 2.33V?
58.	**True**	True or False: In MSP432, the configuration of a single digital I/O pin depends on the values of several registers.
59.	**True**	True or False: In MSP432, one register can affect the configuration of multiple digital I/O pins
60.	**False**	True or False: In MSP432, each digital I/O pin has a dedicated register that controls its configuration.
61.	**False**	True or False: One single port has multiple memory addresses that are associated with different pins.
62.	**True**	True or False: One single port has multiple memory addresses that are associated with different functionalities.

63. **False**	True or False: One single port has one single memory address.
64. **False**	True or False: One single pin has one single memory address.
65. **True**	Consider typical 32-bit and 16-bit microcontrollers: The datapath of a 32-bit microcontroller is twice as wide as a 16-bit microcontroller.
66. **True**	The datapath of a 32-bit microcontroller is twice as wide as a 16-bit microcontroller: The address bus of a 32-bit microcontroller is twice as wide as a 16-bit microcontroller.
67. **True**	The address bus of a 32-bit microcontroller is twice as wide as a 16-bit microcontroller: The Registers of a 32-bit microcontroller have twice as many bits as a 16-bit microcontroller.
68. **True**	The Registers of a 32-bit microcontroller have twice as many bits as a 16-bit microcontroller: The ALU of a 32-bit microcontroller is capable of handling operands that are twice as wide as a 16-bit microcontroller.
69. **False**	The ALU of a 32-bit microcontroller is capable of handling operands that are twice as wide as a 16-bit microcontroller:

		The address space of a 32-bit microcontroller is twice as big of a 16-bit microcontroller.
70.	**False**	True or False: In memory-mapped i/o, peripherals and memory certainly share an address bus.
71.	**False**	True or False: In memory-mapped i/o, peripherals and memory certainly share a data bus.
72.	**True**	True or False: In memory-mapped i/o, peripherals and memory certainly share the same address space.
73.	**True**	True or False: In memory-mapped i/o, memory and the peripherals can share address bus and data bus.
74.	**True**	True or False: In memory-mapped i/o, the processor uses the same instruction for reading from memory and reading from an i/o.
75.	**do not need pull-up or pull-down resistors from within MSP432's GPIO. In other words, PxREN should be equal 0.**	When programming the buttons on the boosterPack, the buttons
76.	**False**	True or False: API is a special form of HAL.
77.	**True**	True or False: HAL is a special form of API.

78. **False**	True or False: HAL functions can be found in the driverlib manual.
79. **True**	True or False: We can use driverlib functions to develop (implement or write the code for) HAL.
80. **False**	True or False: HAL stands for Hardware Abstraction Level.
81. **True**	True or False: Driverlib is a form of API.
82. **False**	True or False: Driverlib is a form of HAL
83. **yes**	Yes or No. It makes the code more readable.
84. **Yes**	Yes or No. It makes the code easier to port from one processor to another.
85. **No**	Yes or No. It reduces the size of the machine code binary after compilation.
86. **No**	Yes or No. It improves the efficiency of the program (e.g. it becomes faster or uses fewer resources.)
87. **True**	True or False: FSM stands for Finite State Machine.
88. **True**	

	True or False: FSM is a mathematical model of computation.
89. **True**	True or False: FSM is an abstract machine.
90. **False**	True or False: FSM is the C code that implements a certain algorithm
91. **False**	True or False: FSM is a mechanical machine that is capable of doing calculations.
92. **False**	True or False: FSM is a machine that can go into an infinite number of states.
93. **True**	True or False: FSM is defined by a list of states, an initial state, and conditions for transitions between the states.
94. **False**	True or False: An FSM is weaker than the combinational logic
95. **True**	True or False: Some examples of the behavior of an FSM are traffic lights, vending machines, and elevators.
96. **False**	True or False: An FSM is more powerful than the Turing
97. **False**	True or False: A static in C is a variable that is constant and does not change.

98. **False**	True or False: A static variable is stored on the stack.
99. **False**	True or False: A static variable defined within function Func1 can be accessed from outside Func1 by other functions.
100. **True**	True or False: A static variable defined within function Func1 holds its value after Func1 finishes. The next time Func1 starts the old value of the static variable is still available.
101. **True**	True or False: Switch debouncing is some form of signal filtering.
102. **True**	True or False: When we press a button, the two metals of the button do not create instant contact, instead, they connect and disconnect several times until they settle in the pressed state and remain connected.
103. **False**	True or False: There is only one method to debounce a button.
104. **False**	True or False: Switch debouncing is some form of signal amplification.
105. **True**	True or False: Switch debouncing creates the illusion of a perfect switch to the user.

106. **True**	True or False: In an FSM, it is possible that some inputs are Don't Care for some of the transition arcs.
107. **True**	True or False: It is possible to use analog components such as resistors and capacitances to debounce a switch.
108. **full-duplex**	What duplex is SPI
109. **half-duplex**	What duplex is I2C
110. **full-duplex**	What duplex is UART
111. **Serial**	SPI, I2C, and UART are serial or non serial?
112. **Yes**	Is SPI synchronous?
113. **Yes**	Is I2C synchronous?
114. **No. Asynchronous**	Is UART synchronous?
115. **5K**	Consider a communication scheme where each symbol (baud) represents two bits. If the bitrate in this communication is 10 Kbit/sec, what is the baudrate?
116. **0.8M**	Consider a communication scheme where the bitrate is 1 Mbit/sec. The bits are packaged as packets of 20 bits, where 4 bits are used to specify the start and stop of the package and other protocol-related information. The rest of the bits are data bit. What is the data rate (in bits per second)?

117. **11**	What are the number of bits in a UART packet of data with the default number of start bits, 8 data bits, 2 stop bits, and no parity bits?
118. **False**	True or False: In UART, sender, and transmitter share a clock signal
119. **True**	True or False: In UART, baud rate is equal to bit rate.
120. **False**	True or False: In UART, bit rate is equal to data rate.
121. **True**	True or False: UART is asynchronous.
122. **False**	True or False: UART is always half-duplex.
123. **True**	True or False: UART has two data lines between sender and transmitter.
124. **true**	Assume we are transmitting single bytes with a parity bit. Choose T/F. When sending ASCII of 'L', the odd parity is 0.
125. **False**	Assume we are transmitting single bytes with a parity bit. Choose T/F. When sending 0x23, the even parity is 0.
126. **True**	Assume we are transmitting single bytes with a parity bit. Choose T/F. When sending the decimal 0, the odd parity is 1.

127. **4 and 0.34**	In a UART where the BRCLK's frequency (BRCLK is the input clock of the UART counter) is 500KHz and Baudrate is 115200, what are the integer part and the fractional part of the division factor (the potential load value of the counter)?
128. **In the middle of the data bit.**	When is the ideal time for sampling the data of a bit?
129. **0.1ms**	A UART module shares a clock with the processor core that is 4MHz. The Baudrate is set at 10,000 baud/sec. What is the length of time the UART has to wait between sampling two consecutive bits?
130. **15 us**	A UART uses Baudrate of 100 K Baud/sec. The counter inside the UART uses a 1MHz clock as its input, how long after the beginning of the start bit (the falling edge moment), we should sample the first data bit?
131. **4 and 0.34**	In a UART where the BRCLK's frequency (BRCLK is the input clock of the UART) is 500KHz and Baudrate is 115200, what are the integer part and the fractional part of the division factor?
132. **1kHz**	What is the clock frequency of a processor whose clock period is 1ms?
133. **2 s**	In terms of seconds, how long is 2 million cycles of a processor with a clock frequency of 1MHz?
134. **3,000,000,000**	

For a processor with a clock frequency of 1 GHz, how many cycles exist in 3 seconds?