Review Test Submission: Test1

User	Malachi A Smith
Course	CE-426-01: Real-Time Embedded Systems - SPRING MWF 11:20-12:20
Test	Test1
Started	5/8/20 11:20 AM
Submitted	5/8/20 11:59 AM
Due Date	5/8/20 12:20 PM
Status	Completed
Attempt Score	100 out of 100 points
Time Elapsed	38 minutes out of 1 hour
Instructions	- The test is timed. You have one hour to complete the test, within the time period when it is released during the test schedule time. - You can use the course resources (lecture slides, exercises, labs, available on Blackboard, and course reference books) - Any form of access to other sources, including electronic sources, such as google or other resources that are not explicitly allowed are not acceptable. - Calculator is allowed. - The test is to be completed independently. Students are not allowed to communicate with anyone during the test.
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

30 out of 30 points

March the terms or phrases on the left to their best description on the right

Question	Correct Match	Selected Match
Foreground-background system	M. An embedded system without an operating system	M. An embedded system without an operating system
Hard real-time system	L. Missing a deadline could result in catastrophic failure	L. Missing a deadline could result in catastrophic failure
Soft real-time system	N.	N.

	The quality of service degrades with missed deadlines, but may not immediately result in total failure	The quality of service degrades with missed deadlines, but may not immediately result in total failure
Thumb instructions number of bits	D. 16-bits	D. 16-bits
Thumb-2 instruction set	J. Combines 16 and 32 bit instructions	J. Combines 16 and 32 bit instructions
Harvard processor architecture	B. Separate instruction and data buses	B. Separate instruction and data buses
Little-endian	C. Multi-byte numbers are ordered in memory from lowest byte to highest byte in increasing memory addresses.	C. Multi-byte numbers are ordered in memory from lowest byte to highest byte in increasing memory addresses.
Link register (LR)	K. Saves the return address during subroutine calls	K. Saves the return address during subroutine calls
The bit widths of the Cortex-M3 CPU registers	E. 32-bits	E. 32-bits
Big-endian	H. Multi-byte numbers are ordered in memory from highest byte to lowest byte in increasing memory addresses.	H. Multi-byte numbers are ordered in memory from highest byte to lowest byte in increasing memory addresses.

All Answer Choices

A. Systems that have deadlines in microseconds

B. Separate instruction and data buses

C.
Multi-byte numbers are ordered in memory from lowest byte to highest byte in increasing memory addresses.

D. 16-bits

E. 32-bits

F. Points to the next instruction to be executed

G. Uses common bus for instructions and data

H.
Multi-byte numbers are ordered in memory from highest byte to lowest byte in increasing memory addresses.

I. Points to the stack during subroutine calls

J. Combines 16 and 32 bit instructions

K. Saves the return address during subroutine calls

L. Missing a deadline could result in catastrophic failure

M. An embedded system without an operating system

N.

The quality of service degrades with missed deadlines, but may not immediately result in total failure

Question 2

5 out of 5 points

If the Cortex-M3 BASEPRI register is set to 0x80, the system will accept an interrupt request that has a priority setting of 0x90.

Selected Answer: ✓ False

Answers: True

✓ False

Question 3

5 out of 5 points

Exceptions in Cortex-M3 are always executed in privileged level.

Selected Answer: ✓ True

Answers: ✓ True

False

Question 4

5 out of 5 points

Which of the following instruction sequences would you use to set the priority and sub-priority of IRQ# 33 to 0x90? Assume the following addresses are given:

SETENA	0xE000E100
CLRENA	0xE000E180
SETPEND	0xE000E200
CLRPEND	0xE000E280
IP	0xE000E400

Selected Answer:

LDR R0, =0xE000E400
MOV R1, #0x90
✓ B. STRB R1, [R0, #33]

Answers:

LDR R0, =0xE000E200
MOV R1, #0x90
STR R1, [R0, #33]
A.

LDR R0, =0xE000E400
MOV R1, #0x90
✓ B. STRB R1, [R0, #33]

LDR R0, =0xE000E400
MOV R1, #0x33
STR R1, [R0, #90]

C.

LDR R0, =0xE000E100
MOV R1, #0x33
STRB R1, [R0, #90]
D.

Question 5

5 out of 5 points

Which of the following instruction sequences would you use to enable IRQ# 30? Assume the following addresses are given:

SETENA	0xE000E100
CLRENA	0xE000E180
SETPEND	0xE000E200
CLRPEND	0xE000E280
IP	0xE000E400

Selected Answer:
LDR R0, =0xE000E100
LDR R1, =0x40000000
✓ B. STR R1, [R0,#0]

Answers:
LDR R0, =0xE000E200
LDR R1, =0x00000000
A. STR R1, [R0,#30]

LDR R0, =0xE000E100
LDR R1, =0x40000000
✓ B. STR R1, [R0,#0]

LDR R0, =0xE000E100
LDR R1, =0x30
C. STR R1, [R0,#0]

LDR R0, =0xE000E200
LDR R1, =0x80000000
D. STR R1, [R0,#0]

Question 6

5 out of 5 points

Which of the following instruction sequences would you use to disable IRQ# 64? Assume the following addresses are given:

SETENA	0xE000E100
CLRENA	0xE000E180
SETPEND	0xE000E200
CLRPEND	0xE000E280
IP	0xE000E400

Selected Answer:
LDR R0, =0xE000E180
LDR R1, =0x00000001
✓ C. STR R1, [R0,#8]

Answers:
LDR R0, =0xE000E180
LDR R1, =0x00000000
A. STR R1, [R0,#64]

LDR R0, =0xE000E180
LDR R1, =0x64
B. STR R1, [R0,#0]

LDR R0, =0xE000E180
LDR R1, =0x00000001
✓ C. STR R1, [R0,#8]

```
    LDR R0, =0xE000E280
    LDR R1, =0x80000000
D. STR R1, [R0,#4]
```

Question 7

20 out of 20 points

For the following program segment assume the stack operations behave in Cortex-M3 default form, i.e. pre-adjust downwards (i.e. pre-decrement) stack. If you set a breakpoint at line number 7 and let the program run from the beginning until the breakpoint, determine the contents of the registers R1, R2, R3, LR, SP, and PC, and the program stops at the breakpoint.

The initial value of the SP register before the program starts execution is 0x20020000.

Assume a little-endian processor. Note that the starting address of each instruction is given as a comment (Assume they are correct).

```
1:        main   LDR R1, =0x00112233          ; 0x080001A8
2:               LDR R2, =0x55667788          ; 0x080001BA
3:               LDR R3, =0x9900AABB          ; 0x080001BC
4:               PUSH {R3}                    ; 0x080001BE
5:               BL       sub1                ; 0x080001B0
6:               POP {R1}                     ; 0x080001B4
7:               B .                          ; 0x080001B6
8:        sub1    push {R2, LR}               ; 0x080001B8
9:               BL       sub2                ; 0x080001CA
10:              pop {R3,PC}                  ; 0x080001CE
11:       sub2    push {R1, LR}               ; 0x080001C0
12:              pop {R2, PC}                 ; 0x080001C2
```

Question	Correct Match	Selected Match
R1	A. 0x9900AABB	A. 0x9900AABB
R2	B. 0x00112233	B. 0x00112233
R3	C. 0x55667788	C. 0x55667788
LR	D. 0x080001CF	D. 0x080001CF
PC	E. 0x080001B6	E. 0x080001B6
SP	F. 0x20020000	F. 0x20020000

All Answer Choices

A. 0x9900AABB

B. 0x00112233

C. 0x55667788

D. 0x080001CF

E. 0x080001B6

F. 0x20020000

G. 0x2001FFFC

H. 0x080001B4

Question 8

5 out of 5 points

What is the bit band alias address for 0x20000500 bit 4? Write your answer as a hex value with the 0x prefix.

Selected Answer: 0x2200A010

Correct Answer:

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	0x2200A010	

Question 9

10 out of 10 points

What is the address and bit number in the bit band region that corresponds to the alias address 0x4200049C? Write the address as hex value with the 0x prefix and the bit number as a decimal value without any prefix or suffix.

Address = **[Address]**

Bit number = **[Bit]**

Specified Answer for: Address 0x40000024

Specified Answer for: Bit 7

Correct Answers for: Address		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	0x40000024	
Correct Answers for: Bit		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	7	

Question 10

10 out of 10 points

Using the Cortex-M3's parameter passing convention, how will the input parameters be passed when a call to the following function is made:

```
void   myFunc(int64_t* array, uint32_t size, uint32_t* answer);
```

Selected Answer:
R0 = array
R1 = size
C. R2 = answer

Answers:
R1:R0 = array
R2 = size
R3 = answer
A.

R3:R2 = array
R1 = size
B. R0 = answer

R0 = array
R1 = size
C. R2 = answer

[stack] = array
R1 = size
D. R2 = answer

Thursday, July 9, 2020 11:54:57 AM EDT

← OK