USN | | | | | | | | | | |

Internal Assessment Test 1 – May 2022

Sub:	Embedded Systems					Sub Code:	18EC62	Branch:	ECE
Date:	06-05-2022	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	6/A,B,C,D		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	With a neat diagram, explain the architecture of ARM Cortex M3 Microcontroller.	[10]	CO1	L1
2	Explain the stack operations using PUSH and POP instructions in ARM Cortex M3.	[10]	CO1	L2
3	Explain the applications of Cortex M3.	[10]	CO1	L2
4	With a neat diagram, explain operation modes and privilege levels in Cortex M3.	[10]	CO1	L2
5	Explain ARM Cortex-M3 Program Status Register in detail.	[10]	CO1	L2
6	Describe the functions of exceptions with a vector table and priorities in cortex m3	[10]	CO1	L2

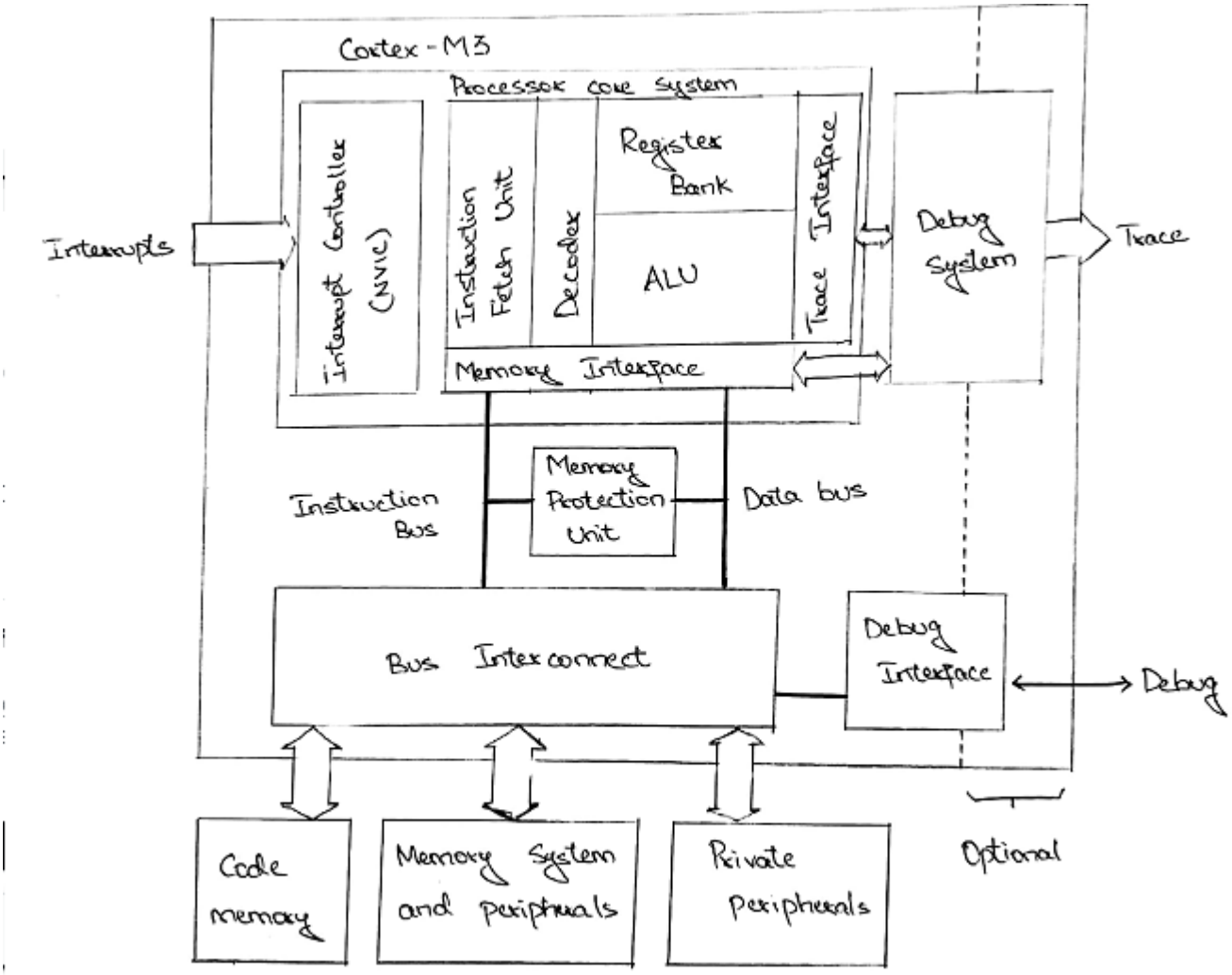
USN | | | | | | | | | | |

Internal Assessment Test 1 – May 2022

Sub:	Embedded Systems					Sub Code:	18EC62	Branch:	ECE
Date:	06-05-2022	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	6/A,B,C,D		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	With a neat diagram, explain the architecture of ARM Cortex M3 Microcontroller.	[10]	CO1	L1
2	Explain the stack operations using PUSH and POP instructions in ARM Cortex M3.	[10]	CO1	L2
3	Explain the applications of Cortex M3.	[10]	CO1	L2
4	With a neat diagram, explain operation modes and privilege levels in Cortex M3.	[10]	CO1	L2
5	Explain ARM Cortex-M3 Program Status Register in detail.	[10]	CO1	L2
6	Describe the functions of exceptions with a vector table and priorities in cortex m3	[10]	CO1	L2

1. With a neat diagram, explain the architecture of ARM Cortex M3 Microcontroller.



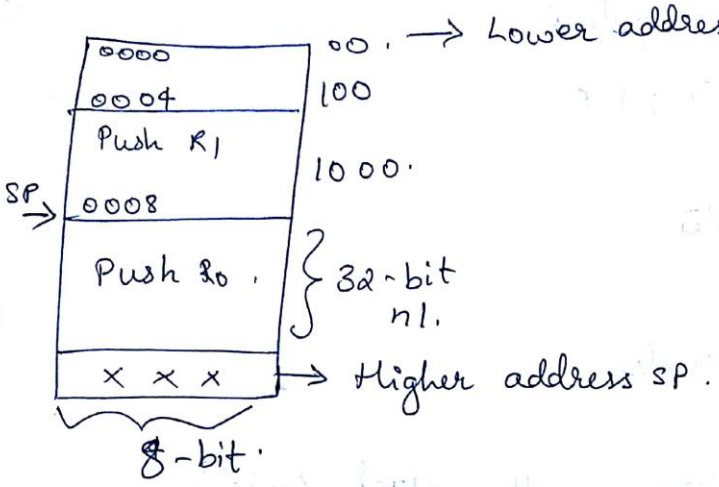
The Cortex-M3 is a 32-bit microprocessor. It has a 32-bit data path, a 32-bit register bank, and 32-bit memory interfaces. The processor has a Harvard architecture, which means that it has a separate instruction bus and data bus. This allows instructions and data accesses to take place at the same time, and as a result of this, the performance of the processor increases because data accesses do not affect the instruction pipeline. This feature results in multiple bus interfaces on Cortex-M3, each with optimized usage and the ability to be used simultaneously.

For complex applications that require more memory system ~~failure~~ features, the Cortex-M3 processor, has an optional Memory Protection Unit (MPU), and it is possible to use an external cache if it's required. Both little endian and big endian memory systems are supported.

The Cortex-M3 processor includes a number of fixed internal debugging components. These components provide debugging operation supports and features, such as breakpoints and watchpoints. In addition, optional components provide debugging features, such as instruction trace, and various types of debugging interfaces.

2. Explain the stack operations using PUSH and POP instructions in ARM Cortex M3.

22. Explain stack memory operations on Cortex M3 processor with necessary diagrams.



Push : Decrements SP by 4.
POP : Increments SP by 4.

Push {R0}; R13 = R13 - 4, then memory [R13] = R0
POP {R0}; R0 = memory [R13], then R13 = R13 + 4

Multiple registers can be pushed & popped in one instruction.

Eg: Push {R0, R1}
Push {R0-R7, R12, R8}

Cortex M3 contains 2 stack pointers (R13). They are banked so that only one is visible at a time.
* Main stack pointer (MSP) : The default stack pointer
* Process stack pointer (PSP) – Used by user application Code.

Name.

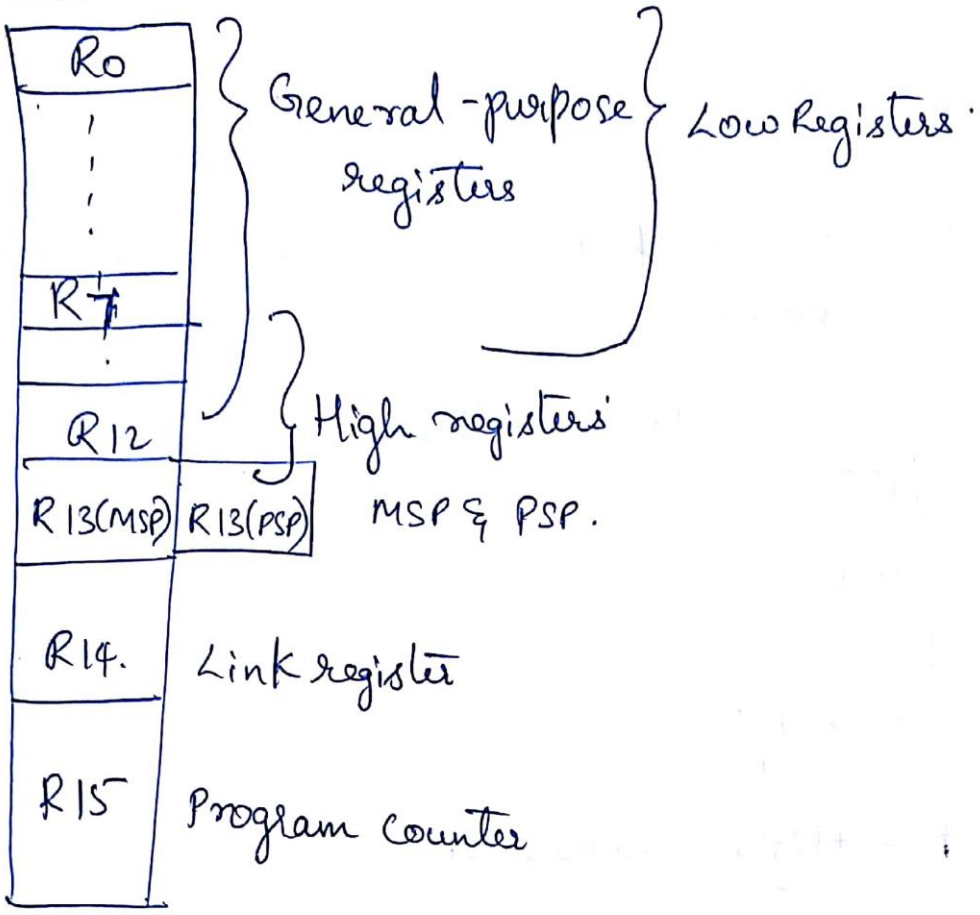


Fig: Diagram showing the stack register.

3. Explain the applications of Cortex M3.

Jeena Jacob
ICRISTEG024

10 List and explain the application of cortex M3 processor.

⇒

(a) Low-cost micro-controller:

↳ It is ideally suited for low-cost micro-controller which are commonly used in consumer products from toys to electrical appliances.

↳ It is highly competitive market due to the many well known 8-bit & 16-bit microcontroller products on the market.

(b) Automotive:

↳ It has very high performance efficiency & low interrupt latency allowing it to be used in real-time system in the automotive industry.

↳ It supports 240 external vectored interrupts with a built-in interrupt controller with nested interrupt supports & an optical MPU making it ideal for highly integrated & cost-sensitive automotive application.

(c) Data communication:

↳ It's low power & high efficiency features coupled with thumb-2 instructions for bit-field manipulation make the CORTEX M3 ideal for many communication application.

(d) Industrial control:

↳ The processors interrupts & features

low latency & enhanced fault-finding features make it a strong candidate in this area.

(e) Consumer products:

↳ CORTEX-M3, being a small processor, is highly efficient & low in power & supply support a MPU enabling complex software to execute while providing robust memory protection.

4. With a neat diagram, explain operation modes and privilege levels in Cortex M3.

(15) List & Explain various operating modes of Cortex M3 processor.

Two types of operating modes:-

(a) Thread mode.

(b) Handler mode.

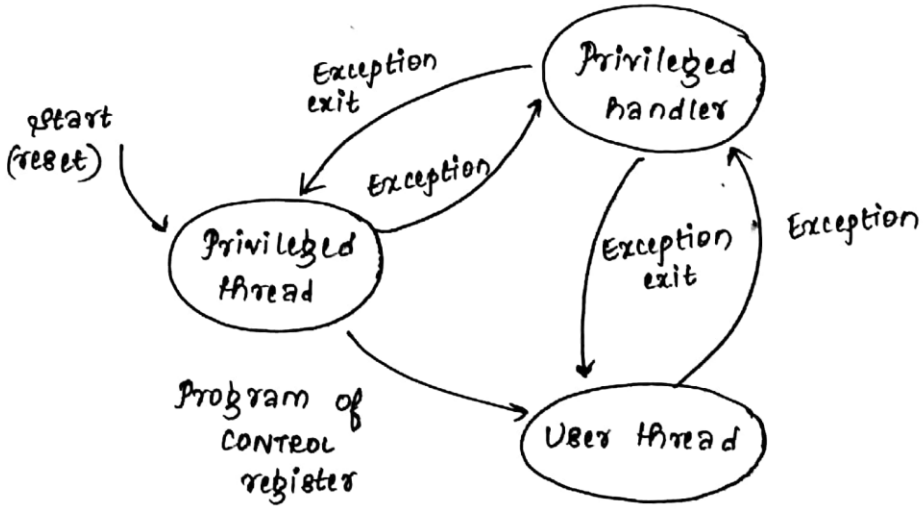
The privilege levels provide a mechanism for safeguarding memory access to critical regions as well as providing a basic security model.

(a) Thread mode:- Used to execute application software.

The processor enters Thread mode when it comes out of reset.

(b) Handler mode:- Used to handle exceptions.

Thread mode & handler mode determine whether the processor is running a normal program or running an exception handler like an interrupt handler or system exception handler.



∴ CONTROL register bits directs modes & levels.

→ Privileged level & user level provide a mechanism for safeguarding memory accesses to critical regions as well as providing a basic security model.

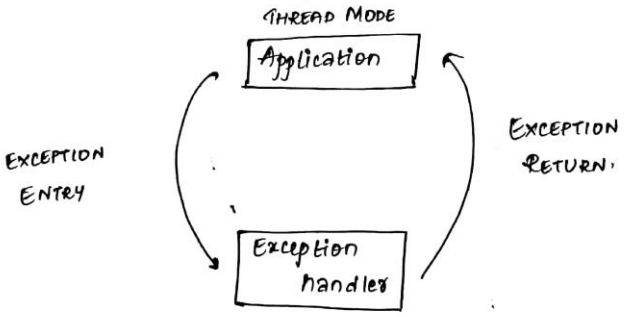
The privilege levels:-

(1) Unprivileged:- The software,

- Has limited access to control register (MSR & MRS) instructions, and cannot use the co-processor (CPS) instruction.
- cannot access the s/m timer, NVIC or system control block.
- Might have restricted access to memory or peripherals.
- Unprivileged software can use the exceptions (SVC) instruction to make a supervisor call to transfer control to privileged software.

(2) Privileged:- The software can use all the instructions & has access to all resources.

- Only privileged software can write to control register.
- In Handler mode, software execution is always privileged.



5. Explain ARM Cortex-M3 Program Status Register in detail.

14. Explain structure of CPSR

Goutham K.
1GR15TED19

A:- The PSR are subdivided into 3 :-
1) APSR 2) IPSR 3) EPSR
They can be accessed together or seperately using special register access MSR and MRS. When accessed as a collective item xPSR is used.

MRS r0, APSR read flag state into R0
MSR APSR, r0 write flag state.

	31	30	29	28	27	26:25	24	23:20	19:16	15:10	9	8 7 6:0
APSR	N	Z	C	V	Q							Exception num.
IPSR												
EPSR						ICI/IT	T			ICI/IT		

Bit	Description
N	Negative
Z	Zero
C	Carry/borrow
V	Overflow
Q	Sticky saturath flag
ICI/IT	Interupt-continuable Instruction, IF-Then instruction status bit
T	Thumb stat
Exception num.	which exception the processor is handling

In Keil assembler, when accessing xPSR is required,
MRS r0, PSR read combined program status word
MSR PSR, r0 write combined status word.

6. Describe the functions of exceptions with a vector table and priorities in cortex m3

Explain interrupts and exceptions in Cortex M3 processor.

Exceptions

Exceptions are numbered 1 to 15 for system exceptions and the rest 240 for external interrupt inputs (Total 256 entries in vector table)

Most of the exceptions have programmable priority, and a few have fixed priority.

The value of the current running exception is indicated by the special register IPSR or from the NVIC's Interrupt Control state Register.

Exception Number	Exception Type	Priority	Description
1.	Reset	-3 (Highest	Reset
2	NMI	-2	Nonmaskable interrupt (external NMI i/p)
3	Hardfault	-1	All fault condition, if the corresponding fault handler is not enabled.
4.	MemManage fault	Programmable	Memory management fault, MPU violation access to illegal locations
5.	Bus fault	Programmable	Bus error like Prefetch abort
6.	Usage fault	Programmable	Exceptions due to program error or typing to access coprocessor.

7-10	Reserved	N/A	—
11	SVCall	Programmable	System Service call
12.	Debug Monitor	Programmable	Debug Monitor.
13	Reserved	N/A	—
14	PendSV	Programmable	Pendable request for System device
15.	SYSTICK	Programmable	System Tick Timer
16.	External Interrupt #0	Programmable	External Interrupt
17.	External Interrupt #1	Programmable	External Interrupt
...	...	...	...
255	External Interrupt #239	Programmable	External Interrupt.

Vector Table

The processor will need to locate the starting address of the exception handler when an exception is being handled. This information is stored in the vector table.

Exception Vector Table After Power Up.

Address	Exception Number	Value (Word Size)
0x00000000	—	MSP Initial Value
0x00000004	1	Reset vector (Program counter initial value)
0x00000008	2	NMI handler starting address
0x0000000C	3	Hard fault handler starting address
	...	other handler starting address

For example, if the reset is exception type 1, the address of the reset vector is 1 times 4 (each word is 4 bytes), which equals 0x00000004, and NMI vector (type 2) is located in 2x4 = 0x00000008. The address 0x00000000 is used to store the starting value for the MSP.