Ryerson University

COE 718

LAB & PROJECT SUMMARIES

LAB 1

Introduction to µVision and ARM Cortex M3

Marks Break-down

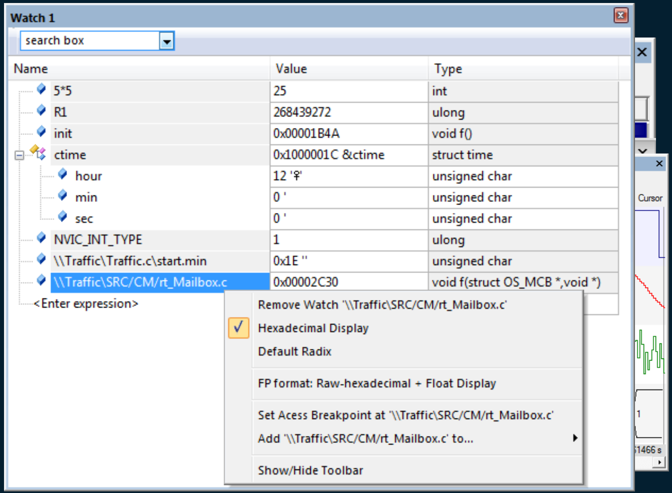
Category	Marks
Demo	25
Code	25
Total	50

µVision IDE

Integrated Development Environment (IDE) for developing software/firmware in C/C++.

TOOLS/FEATURES

- Performance Analyzer
- Execution Profiling
- Logic Analyzer
- Register Window
- Watch Window



Watch 1

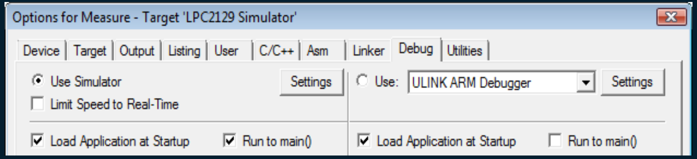
search box

Name	Value	Type
5*5	25	int
R1	268439272	ulong
init	0x00001B4A	void f()
ctime	0x1000001C &ctime	struct time
hour	12 '♀'	unsigned char
min	0 '	unsigned char
sec	0 '	unsigned char
NVIC_INT_TYPE	1	ulong
\\Traffic\Traffic.c\start.min	0x1E ''	unsigned char
\\Traffic\SRC/CM/rt_Mailbox.c	0x00002C30	void f(struct OS_MCB *,void *)
<Enter expression>		

Remove Watch '\\Traffic\SRC/CM/rt_Mailbox.c'
✓ Hexadecimal Display
Default Radix
FP format: Raw-hexadecimal + Float Display
Set Acess Breakpoint at '\\Traffic\SRC/CM/rt_Mailbox.c'
Add '\\Traffic\SRC/CM/rt_Mailbox.c' to...
Show/Hide Toolbar
Cursor
1
1466 s

MODES

- Debug Mode
  1. Use Simulator
  2. Use ULINK ARM Debugger



Options for Measure - Target 'LPC2129 Simulator'

● Use Simulator

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

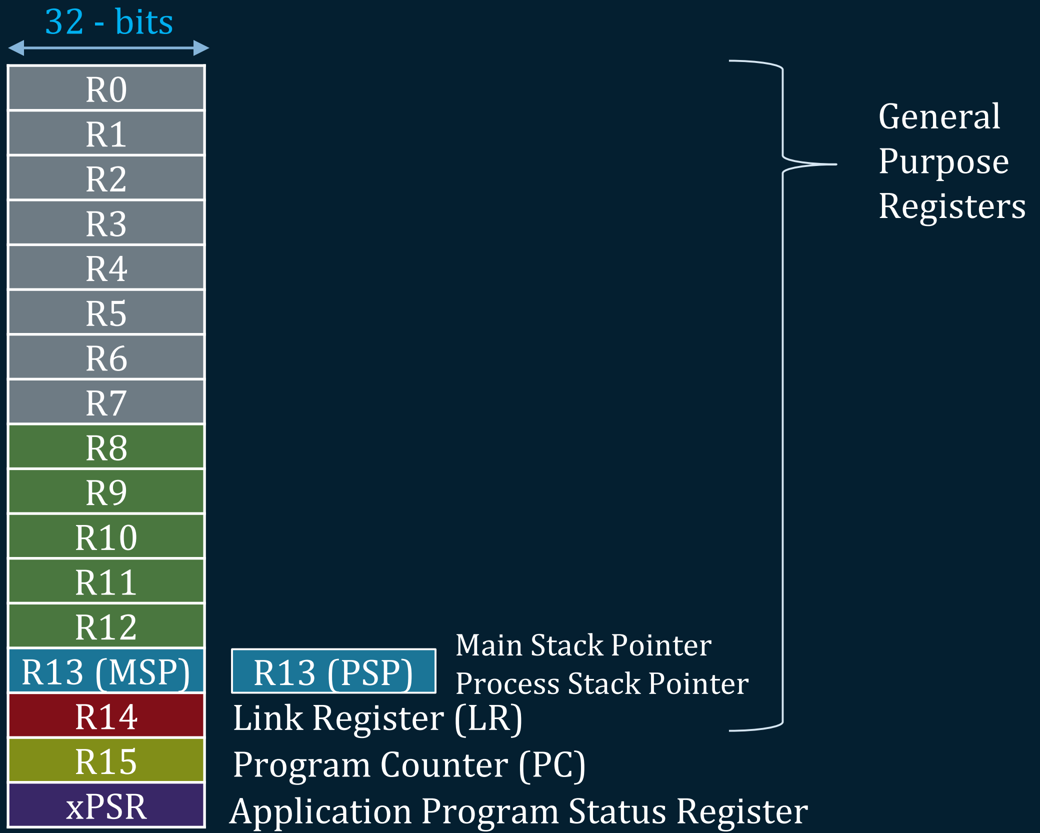
○ Use: ULINK ARM Debugger

☑ Load Application at Startup □ Run to main()

- Normal Mode

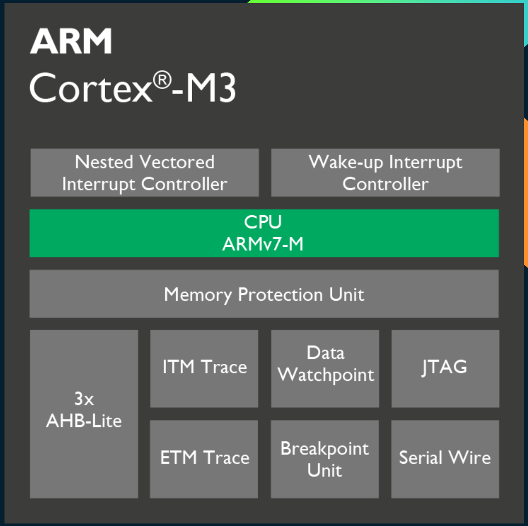
ARM Cortex M3

Advanced RISC Machine (ARM) for embedded applications (M3).



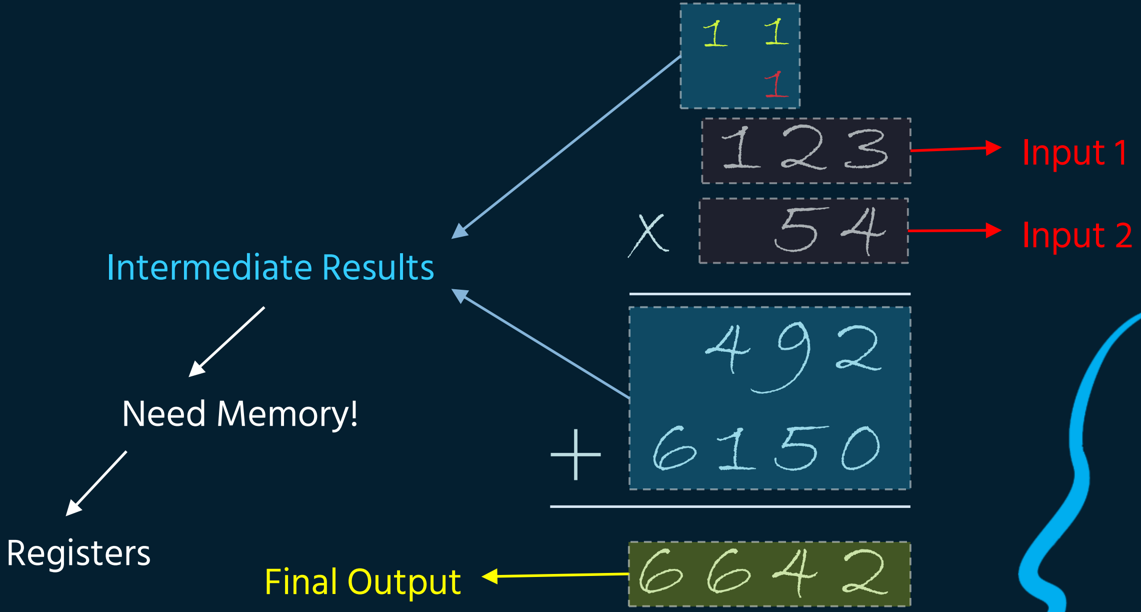
▨ Low Registers

🟩 High Registers



RISC Machines

Reduced Instruction Set Computing Machines: Computing machines which need to be fed with instructions that are broken down into very basic set of operations.



Input 1

Input 2

Intermediate Results

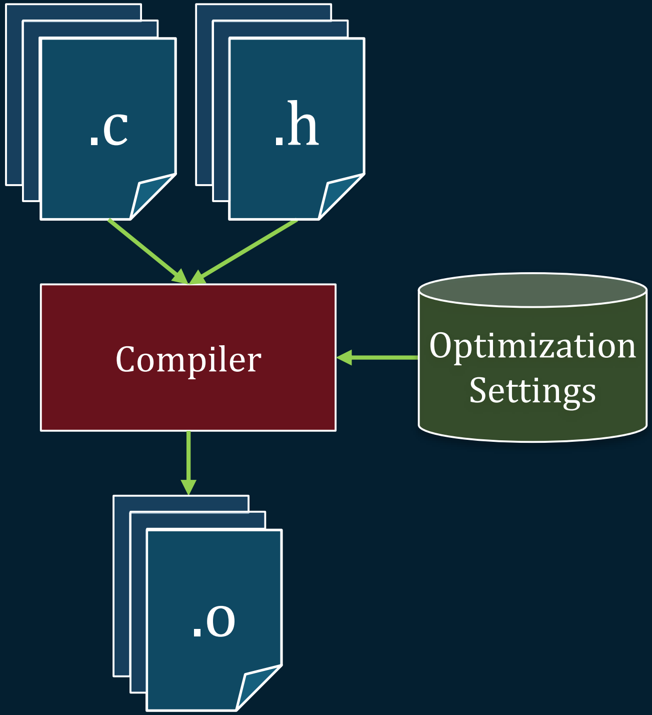
Need Memory!

Registers

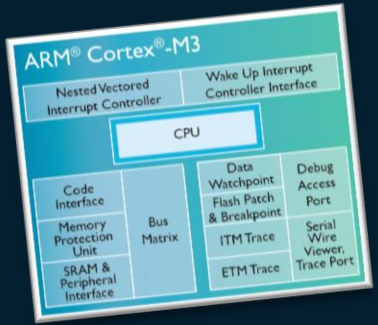
Final Output

Compiler

Turns your C/C++ code into assembly while optimizing speed, memory, and performance.



Default	Use the compiler default .
Level 0 (-O0)	Turn off almost all optimization.
Level 1 (-O1)	Turn off optimizations that seriously degrade the debug view.
Level 2 (-O2)	High optimization (default level).
Level 3 (-O3)	Maximum optimization. Note that Level 3 in combination with Optimize for Time may generate more code that Level 2 since it may unroll loops.



LAB 2

Exploring ARM Cortex M3 Features

Marks Break-down

Category	Marks
Demo	20
Code	50
Report	30
Total	100

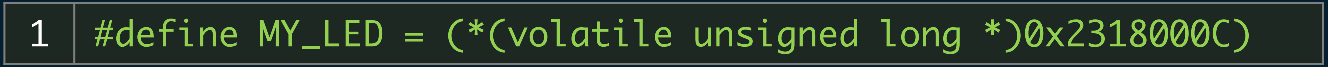
Bit Banding

Allows individual bits in the SRAM and peripheral registers to be read and written to instead of reading a whole register and masking the desired bits.

STEP 1: Calculate the bit address

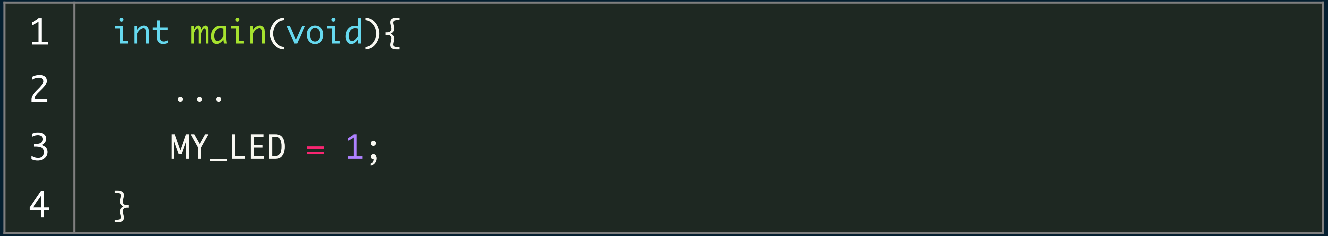
- **STEP A** Calculate Byte Offset
- **STEP B** Calculate Bit Band Word Address

STEP 2: Define a Pointer to the address



```
#define MY_LED = (*(volatile unsigned long *)0x2318000C)
```

STEP 3: Assign a Value to the Port Bit



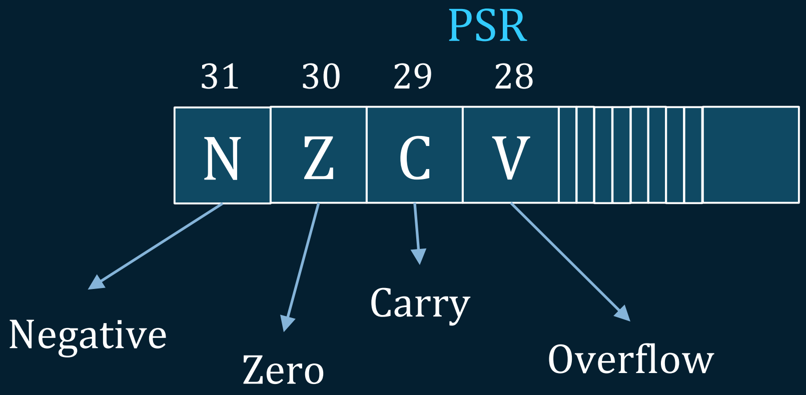
```
int main(void){
   ...
   MY_LED = 1;
}
```

Conditional Execution

ARM allows some instructions to contains conditions within their opcode.

C Code:

```
if(a <= 5)
    a = 10;
else a = 1;
```



Conditional Method:

```
        CMP    R2, #5;
        MOVLE  R2, #10;
        MOVGT  R2, #1;
```

Non-Conditional Method:

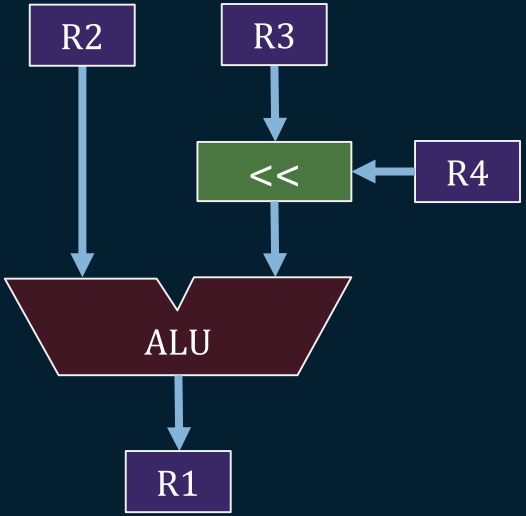
```
        CMP    R2, #5;
        BGT    t_else;
        MOV    R2, #10;
t_else: MOV    R2, #1;
```

Barrel Shifting

Allows shift/rotate the operand before it enters the ALU.

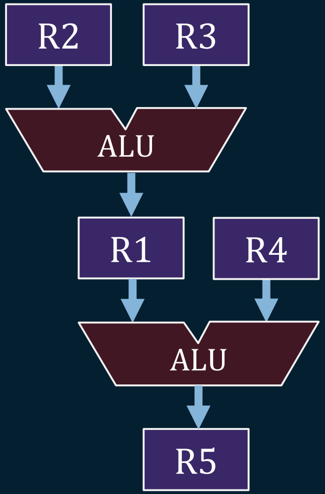
With Barrel Shifting:

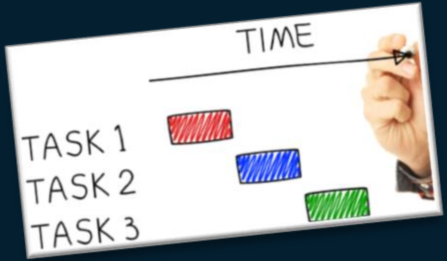
```
ADD   R1, R2, R3 LSL R4;
```



Without Barrel Shifting:

```
MUL   R1, R2, R3;
ADD   R5, R1, R4;
```





LAB 3a

Task Scheduling

Marks Break-down

Category	Marks
Demo	20
Round Robin Scheduling	20
Preemptive Scheduling	35
Non Preemptive Scheduling	25
Total	100

RTX

RTX is a Real-Time Operating System (RTOS) designed for ARM and Cortex-M devices.

SCHEDULER
Allows and manages execution of multiple tasks.

MUTEX
Locks access to critical areas of program. It allows sharing of the same resource, such as file access, but not simultaneously.

EVENT & SEMAPHORE
Software interrupts (events) and semaphore for synchronization.

MAILBOX
Allows message passing between tasks for data exchange or task synchronization.

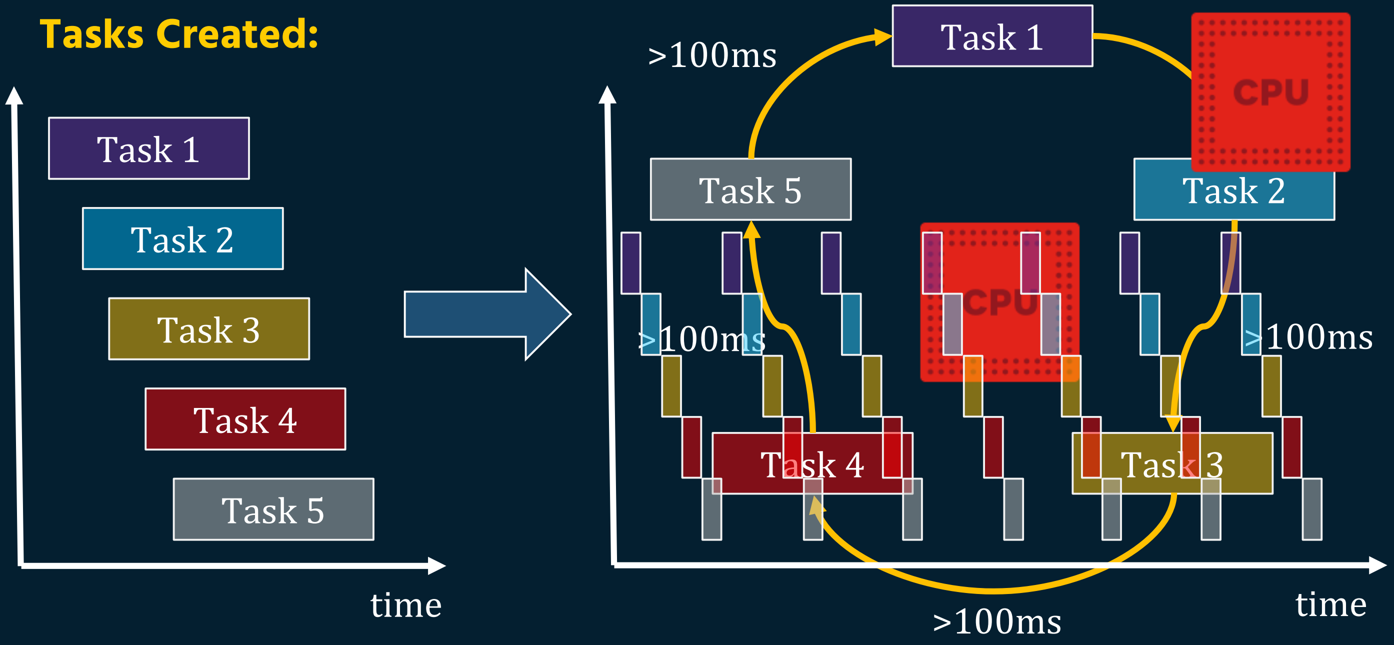
DELAY & INTERVAL
Accurate delay & Interval functions.

MEMORY POOL
Fixed-size blocks of memory that are thread-safe. They operate much faster than the dynamically allocated heap and do not suffer from fragmentation. Being thread-safe, they can be accessed from threads and ISRs alike.

Round-Robin Scheduling

This technique divides processor time equaly into threads ready for execution.

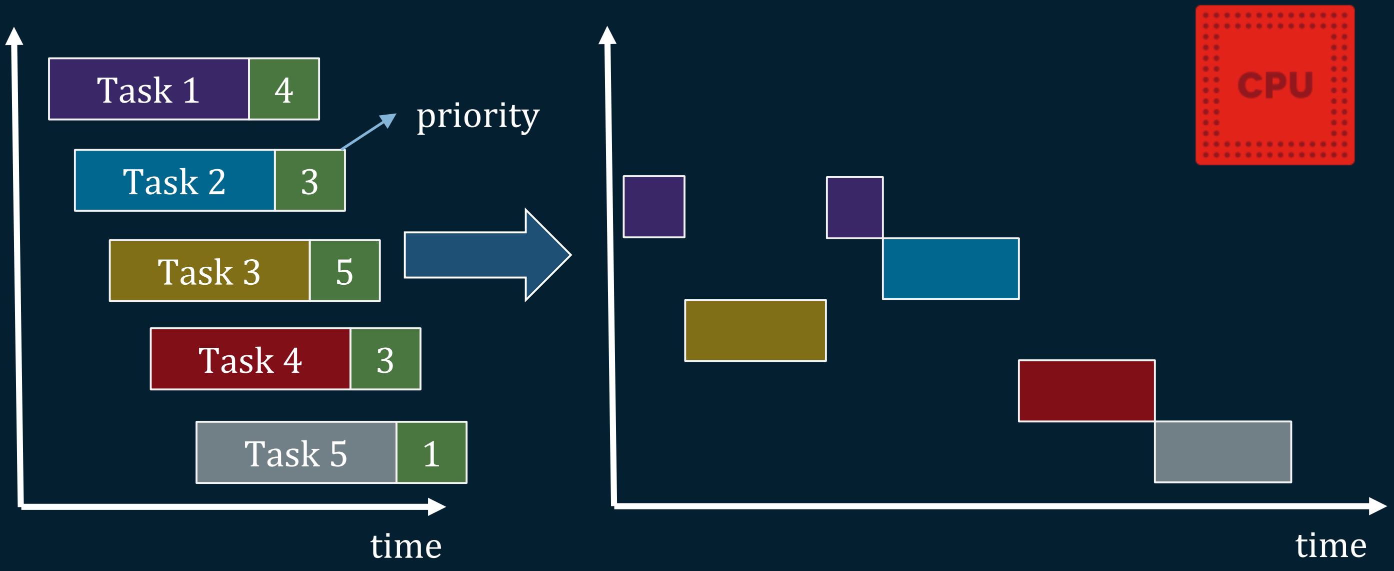
Tasks Created:



Non Pre-emptive Scheduling

Also known as priority-based uninterrupted scheduling

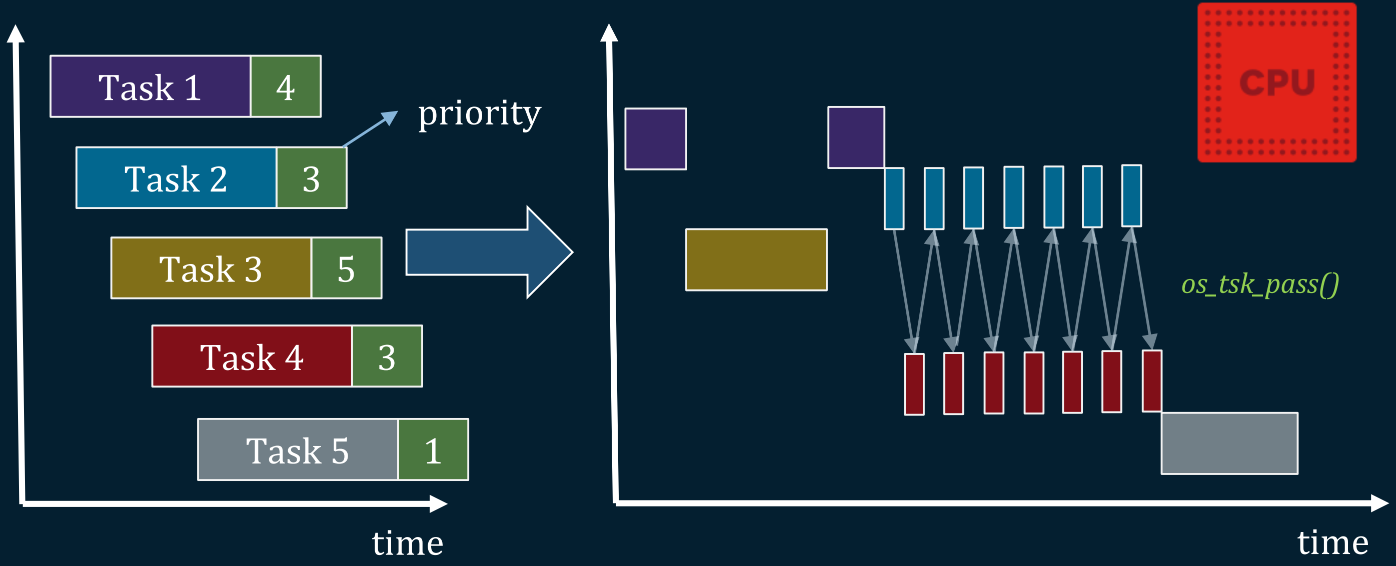
Tasks Created:



Pre-emptive Scheduling

Preempt: To take place of
Threads with same priority share CPU by allowing other thread to take it's place.

Tasks Created:



LAB 3b

Multithreading with RTX

Marks Break-down

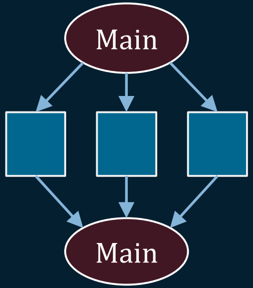
Category	Marks
Demo	20
Report	10
Round Robin Scheduling	20
Preemptive Scheduling	30
Non Preemptive Scheduling	20
Total	100

Thread vs Task

The distinction between a thread and a task is subtle: it is more related to the purpose. A separate thread is usually thought of performing some operation in parallel, usually with the intention of the thread *joining* the parent again; while a task is a separate and parallel sequence of execution without an intention of joining with the parent.

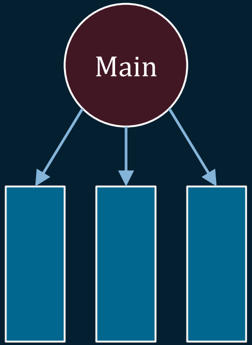
THREAD

- STEP 1 – FORK: Create threads from a common context .
- STEP 2 – EXECUTE: Let threads run in parallel.
- STEP 3– JOIN: After finish execution, gather data from each thread into one context.



TASK

- STEP 1 – INITIALIZE: Set initial variables and parameters.
- STEP 2 – EXECUTE: Let tasks run in parallel.



LAB 4

Real-time Scheduling

Marks Break-down

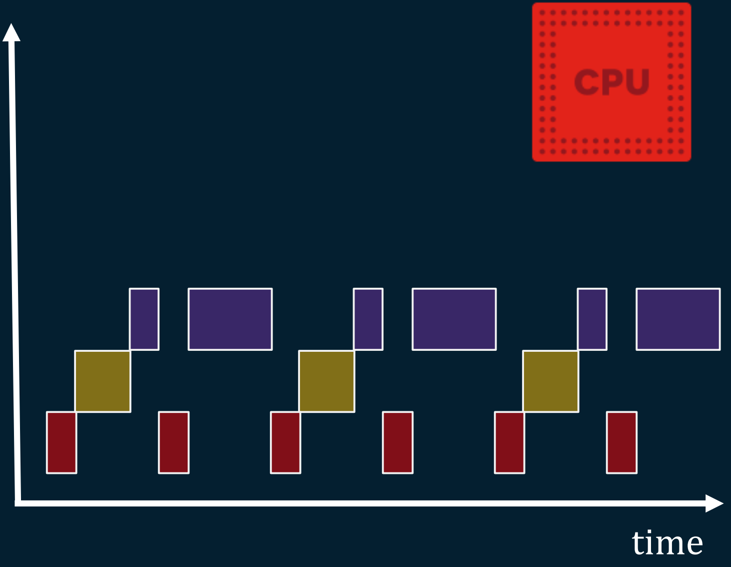
Category	Marks
Demo	20
Question 1	45
Question 2	35
Total	100

Rate-monotonic Scheduling

Higher priorities are assigned to frequently occurring tasks.
Higher rate = high priority.

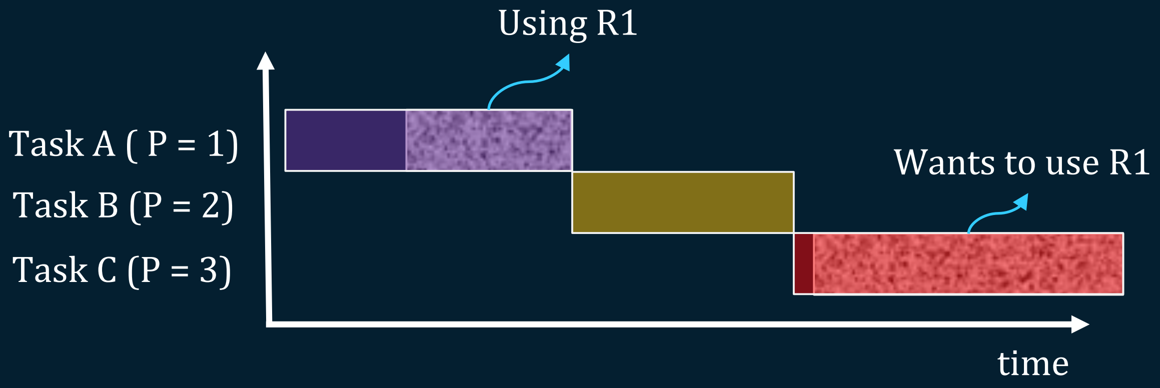
Tasks:

Task	Period (T)	Computation Time (C)	Priority (P)
Task 1	8	4	1
Task 2	8	2	2
Task 3	4	1	3



Priority Inversion

To lower a priority of a high priority task in case of data or resource dependency.



PROJECT

Media Center

Requirements

- A Photo Gallery
- A mp3 Player
- Game(s)
- Any additional Stuff (Sprites, Animations, Apps etc.)

Submission Break-down

Category	Marks	Due Date
Project Summary Report	5	Week 6 (Week of Oct. 10)
Progress Demo	10	Week 9 (Week of Oct. 31)
Interim Report	20	7th November 2016 (Start of Week 10)
Final Demo	30	Week 11 (Week of Nov. 14)
Final Report	25	Week 13 (Week of Nov. 28)
Code	10	

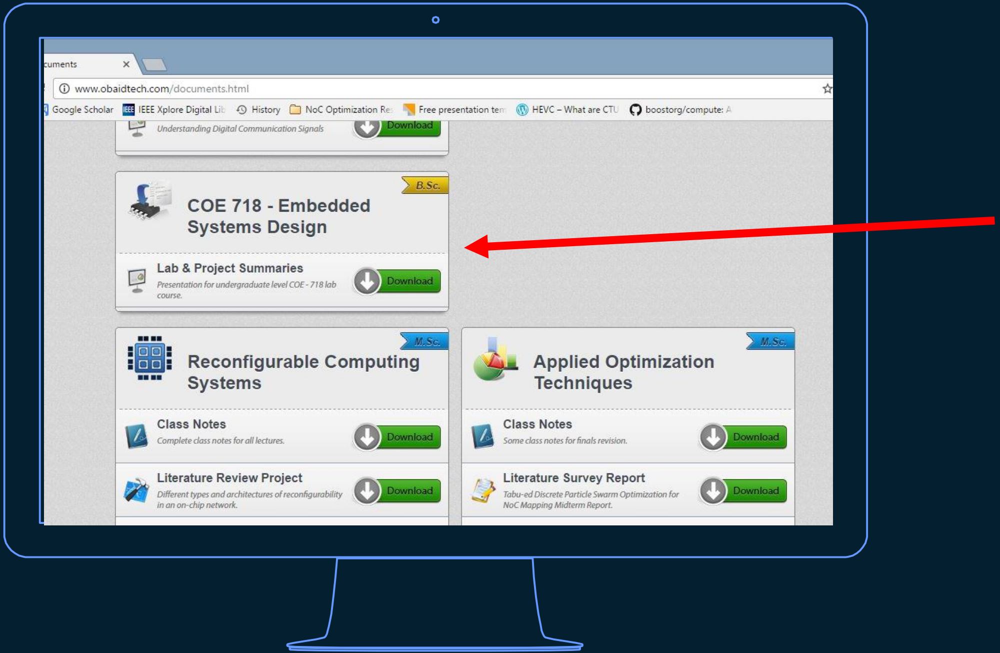
Exact Date

Bonus Projects (2-4% bonus marks)

1. Study of MPEG video file format and development of an MPEG decoder embedded software for an ARM Cortex M3 CPU based MCB1700 den board.
2. Study of MP3 audio encoding and propose a software solution. Then implement your solution with ARM-Cortex M3 (MCB1700) board based solution.
3. Model and implement a suitable hardware-software design for a standard JPEG file encoder/decoder for color images by using a Cyclone IV FPGA based DE2 board for implementation.
4. Design and implement an embedded system suitable for a (student) proposed embedded application. The system may consist of ARM-Cortex M3 (MCB1700 Dev Board), NIOS-II CPU (DE2 board), memory, serial interface, parallel interface for LCD, etc.
5. Configure a typical embedded computer system on the DE2 board or MCB1700 and then implement a real-time application based on an RTOS such as RTX.

Everything we discussed today is available at:

http://www.obaidtech.com/documents.html



Understanding Digital Communication Signals Download

COE 718 - Embedded Systems Design

Lab & Project Summaries
Presentation for undergraduate level COE - 718 lab course. Download

Reconfigurable Computing Systems

Class Notes
Complete class notes for all lectures. Download

Literature Review Project
Different types and architectures of reconfigurability in an on-chip network. Download

Applied Optimization Techniques

Class Notes
Some class notes for finals revision. Download

Literature Survey Report
Tabu-ed Discrete Particle Swarm Optimization for NoC Mapping Midterm Report. Download

THANKS!

Any questions?

You can find me at:
ENG – 402 · mobaidullah@ryerson.ca