CECS346
Midterm Study Guide

Study Materials

Lecture 1 – 4, Chapter 3, 4, 6, Labs 1- 3, Project 1, example projects in TExaSware: C2_Toggle_PF1, C6_InputOutput, C7_SOS, C8_LED, C8_Security, C8_Switch, C8_SwitchLED, C10_SysTick, C10_TableTrafficLight, C10_VendingMachine C10_Odd1sDetector, example project in Labware: Lab2_HelloLaunchPad.

Test Format

There will be three types of questions: multiple choices, true/false, and short answer questions for midterm 1.

Topics

1. ARM Cortex M4 Architecture: Lecture 1, Textbook Chapter 3, Section 3.1
2. TM4C123 GPIO: Lecture 2, Textbook Chapter 2, Section 2.7, Chapter 4, Section 4.1 – 4.2, example projects in TExaSware: C2_Toggle_PF1, C6_InputOutput, C7_SOS, C8_LED, C8_Security, C8_Switch, C8_SwitchLED, example project in Labware: Lab2_HelloLaunchPad.
3. SysTick Timer: Lecture 3, Textbook Chapter 4, Section 4.4, example projects in TExaSware: C10_SysTick.
4. Finite State Machine: Lecture 4, Textbook Chapter 6, Section 6.1 – 6.5, example projects in TExaSware: C10_TableTrafficLight, C10_VendingMachine C10_Odd1sDetector.

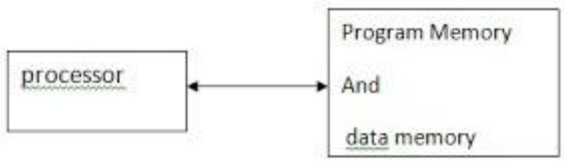
❖ ***SUMMARY OF LECTURES – RECOMMEND GOING THROUGH THE NOTES AGAIN***

Lecture 1: ARM Cortex M4 Architecture

1. Compare Harvard architecture and Von Neumann architecture:

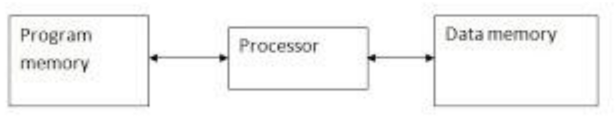
There are basically two types of digital computer architectures. The first one is called Von Neumann architecture and later Harvard architecture was adopted for designing digital computers.

<u>Von Neumann Architecture:</u>



- It is named after the mathematician and early computer scientist John Von Neumann.
- The computer has single storage system (memory) for storing data as well as program to be executed.
- Processor needs two clock cycles to complete an instruction. Pipelining the instructions is not possible with this architecture.
- In the first clock cycle the processor gets the instruction from memory and decodes it. In the next clock cycle the required data is taken from memory. For each instruction this cycle repeats and hence needs two cycles to complete an instruction.
- This is a relatively older architecture and was replaced by Harvard architecture.

Harvard Architecture:

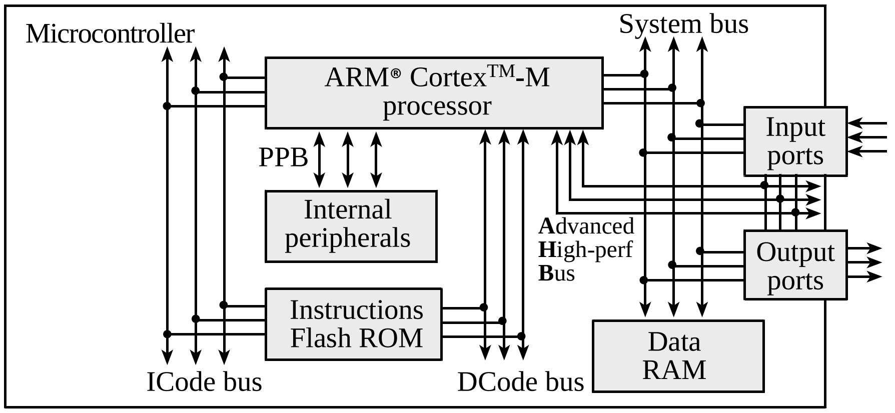


- The name is originated from "Harvard Mark I" a relay based old computer.
- The computer has two separate memories for storing data and program.
- Processor can complete an instruction in one cycle if appropriate pipelining strategies are implemented.
- In the first stage of pipeline the instruction to be executed can be taken from program memory. In the second stage of pipeline data is taken from the data memory using the decoded instruction or address.
- Most of the modern computing architectures are based on Harvard architecture. But the number of stages in the pipeline varies from system to system.

2. System Buses in ARM Cortex M4:

   Data are exchanged with memory and I/O via the system bus interface.

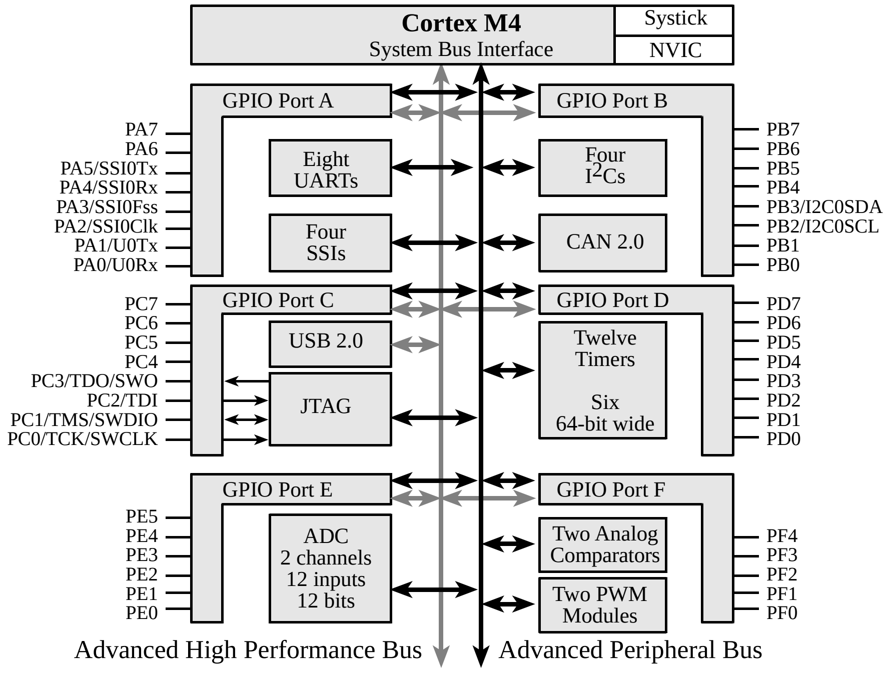
   1) PPB: Private Peripheral Bus
      Used for communication between processor and internal peripherals, such as NVIC (nested vectored interrupt controller)
   2) APB: Advanced Peripheral Bus
      Used for communication between processor and external peripherals
   3) AHB: Advanced High –Performance Bus
      Used for high speed external devices like USB



3. Explain memory-mapped processor architecture: I/O devices are connected like memory; I/O devices are assigned addresses; software accesses I/O using these addresses with the same instructions as accessing a memory location.

Lecture 2 - TM4C123 GPIO

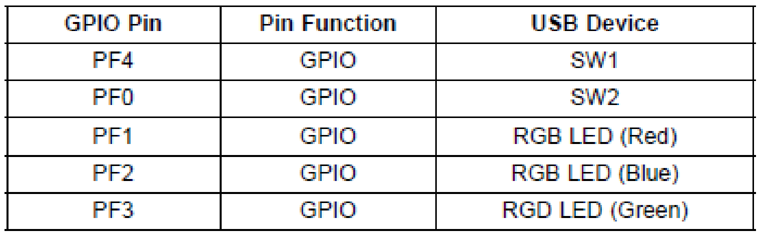
1. GPIO ports in TM4C123: there are 6 General-Purpose I/O (GPIO) ports: Four 8-bit ports (A, B, C, D), One 6-bit port (E), One 5-bit port (F). All of them can be used for two possibilities: digital I/O or an alternative function.



2. Some alternative functions GPIO can be used to provide: UART, SSI (SPI), I^{2}C, Timer, PWM, ADC, Analog, QEI, USB, Ethernet, CAN, etc.
3. GPIO pins can be assigned to as many as **8** different I/O functions; configured for digital I/O, analog input, timer I/O, or serial I/O. Two buses used for I/O: AHPB & APB, digital I/O ports are connected to both.

4. LaunchPad I/O connections for onboard switches and LEDs:

Table 2-2. User Switches and RGB LED Signals



GPIO Pin	Pin Function	USB Device
PF4	GPIO	SW1
PF0	GPIO	SW2
PF1	GPIO	RGB LED (Red)
PF2	GPIO	RGB LED (Blue)
PF3	GPIO	RGD LED (Green)

5. Registers used to control GPIO

Register Name: GPIO_PORTx_name_R	**Functionality**
PCTL	Used for digital functions: each pin has 4 bits set to specify the alternative function for that pin (0 means regular I/O port). See Table 4.3 in textbook.
AMSEL	Analog mode select: 0/1 to disable/enable analog
DEN	0/1 to disable/enable digital
AFSEL	0/1 to disable/select alternate function
PUR	0/1 disable/enable pull resistors
DIR	Direction register: 0 for input, 1 for output
CR	Commit register: determines which bits of the AFSEL, PUR, and DEN registers are committed when a write to these registers is performed.
DATA	Data register, hold data to be transferred or received. It is used to perform the actual input/output on the port.
RCGCGPIO (Run Mode Clock Gating Control)	Provides software the capability to enable and disable GPIO modules in Run mode. When enabled, a module is provided a clock and accesses to module registers are allowed. When disabled, the clock is disabled to save power and accesses to module registers generate a bus fault. To support legacy software, the **RCGC2** register is available. A write to the **RCGC2** register also writes the corresponding bit in this register.
LOCK	Used to unlock certain ports

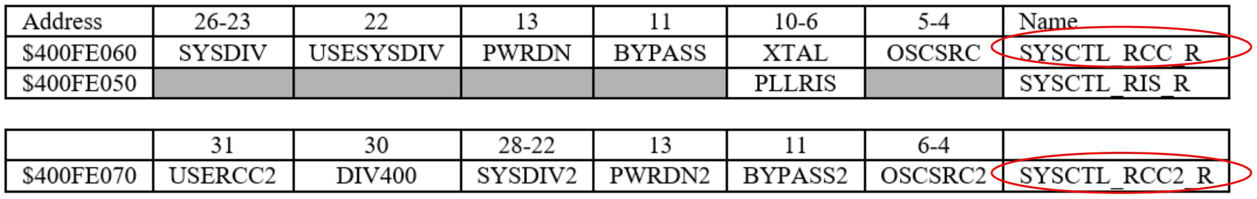
* See TM4C123 data sheet: Chapter 10.5 for detail.

6. Initialization (executed once at beginning)

    1) Turn on Port F clock in **SYSCTL_RCGCGPIO_R/ SYSCTL_RCGC2_R** Wait two bus cycles (two **NOP**)
    2) Unlock PF0 (PD7 also needs unlocking)
    3) Clear *AMSEL* to disable analog
    4) Clear *PCTL* to select GPIO
    5) Set *DIR* to 0 for input, 1 for output
    6) Clear *AFSEL* bits to 0 to select regular I/O
    7) Set *PUR* bits to 1 to enable internal pull-up
    8) Set *DEN* bits to 1 to enable data pins
7. LED and switch connections: positive logic vs. negative logic.
8. Know how to use bit-wise operators to write user-friendly code.
9. Example projects:
    1) C2_Toggle_PF1: Simple I/O with port F one pin
    2) C6_InputOutput: Simple I/O with port F
    3) C7_SOS: Port F control onboard LEDs and SW
    4) C8_LED: Use GPIO Port A to control LED
    5) C8_Switch: Use port F control onboard leds and switches.
    6) C8_SwitchLED: PD3 is an output to LED output with positive logic; PD0 is an input from switch with negative logic
    7) C8_Security: Use GPIO Port E to build a simple security system
10. Questions from the textbook: Chapter 4, Exercise 4.1, 4.3 – 4.7, 4.10 – 4.11

Lecture 3 - SysTick Timer

1. Clock in TM4C123: one internal oscillator and an external oscillator, both are 16MHz. By default, the internal oscillator is used. The internal oscillator with frequency 16MHz ±1% is significantly less precise than the crystal, but it requires less power and does not need an external crystal.



Address	26-23	22	13	11	10-6	5-4	Name
$400FE060	SYSDIV	USESYSDIV	PWRDN	BYPASS	XTAL	OSCSRC	SYSCTL_RCC_R
$400FE050	▨	▨	▨	▨	PLLRIS	▨	SYSCTL_RIS_R

	31	30	28-22	13	11	6-4	
$400FE070	USERCC2	DIV400	SYSDIV2	PWRDN2	BYPASS2	OSCSRC2	SYSCTL_RCC2_R

Table 4.7b. Main clock registers for TM4C123

XTAL	*Crystal Freq (MHz)*
0x0	Reserved
0x1	Reserved
0x2	Reserved
0x3	Reserved
0x4	3.579545 MHz
0x5	3.6864 MHz
0x6	4 MHz
0x7	4.096 MHz
0x8	4.9152 MHz
0x9	5 MHz
0xA	5.12 MHz
0xB	6 MHz (reset value)
0xC	6.144 MHz
0xD	7.3728 MHz
0xE	8 MHz
0xF	8.192 MHz
0x10	10.0 MHz
0x11	12.0 MHz
0x12	12.288 MHz
0x13	13.56 MHz
0x14	14.31818 MHz
0x15	16.0 MHz
0x16	16.384 MHz
0x17	18.0 MHz
0x18	20.0 MHz
0x19	24.0 MHz
0x1A	25.0 MHz
0x1B	Reserved
0x1C	Reserved
0x1D	Reserved
0x1E	Reserved
0x1F	Reserved

Table 4.7a. XTAL field used in the SYSCTL_RCC_R register of the TM4C123

2. Understand how systick timer works: initialize the timer and use it to generate required timing.
   Example:
   Assume 16MHz clock is used, calculate the RELOAD value for SysTick so that we can use it to generate a 50Hz squarewave.
   Clock duration Δt=1/16MHz=62.5 ns
   Half of a square wave = 1/50/2=10ms,
   RELOAD value=10ms/62.5ns-1=160,000-1=159999 <2^{24}=16,777,216

   Initialization SysTick

   1) Clear ENABLE to stop counter
   2) Update RELOAD value
   3) Clear the counter via NVIC_ST_CURRENT_R
   4) Set CLK_SRC=1 and specify interrupt action via INTEN in NVIC_ST_CTRL_R

Address	31-24	23-17	16	15-3	2	1	0	Name
$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
$E000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
$E000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R

SysTick Registers

3. Questions from the textbook: Chapter 4, Exercise 4.14, 4.16, D4.17 – D4.19
   4.14, 4.16: hint: the key is to calculate the RELOAD value for SysTick.
   D4.17 – D4.19: design problem similar to Lab 2, with different logic to control the LEDs.

Lecture 4 – Finite State Machine, Pointers, Arrays, and Structures

1. Know how to declare, define, and use arrays and structures
2. Know how to design an embedded system using Moore FSM.
3. Review traffic controller example table implementation.
4. Example projects to review:
   C10_TableTrafficLight
   C10_VendingMachine
   C10_Odd1sDetector
5. Questions from the textbook: Chapter 6, Exercise D6.14, answer the question in C.

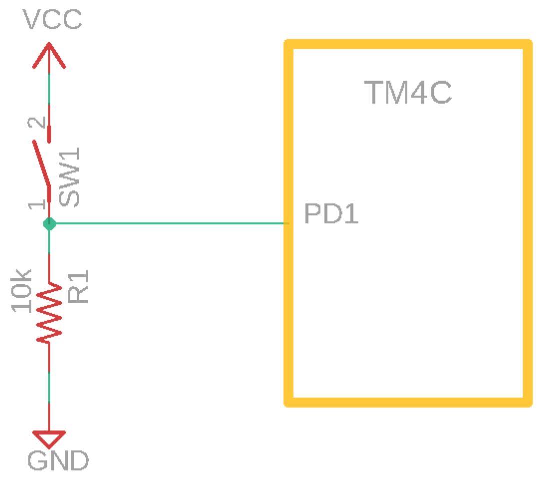
Practice Problems:

1. Assume we want to use Port D pins 0 & 1 as inputs, and pin 2 & 3 as outputs. Do you need to use the unlock register to use these pins in Port D?
2. Assume we want to use Port D pins 0 & 1 as inputs, and pin 2 & 3 as outputs. Write the line(s) of code using "friendly" initialization to set the direction register for Port D to accomplish this.
3. Assume we want to use Port D pins 0 & 1 as inputs, and pin 2 & 3 as outputs. Write the line(s) of code using "friendly" initialization to set the digital enable register for Port D for this configuration.
4. Assume we are using Port D pins 0 & 1 as inputs. What is the bit specific address that we could use to retrieve only those two pin values? (Answer can be in the form of multiple numbers in any base (hex, decimal, etc) with addition/subtraction/multiplication)
5. What is the last step you must do to use a GPIO port for digital I/O?

6. Assume variable `b` is initialized using `b |= 0x95`. The value of `b` after executing `b ^= 0x27` is _________. (Show your work)

7. Assume variable `b` is initialized using `b = 0x95`. The value of `b` after executing `b ^= 0x27` is _________. (Show your work)

8. Given the below figure, is switch SW1 considered positive or negative logic? _______



9. What value has the 8-bit number 0x70 when converted to decimal and binary representations?

10. The intent of the function Fun1 is to call another function Output with the values from 1 to 100. There is one bug. Fix the bug in the code by adding, changing or removing as necessary

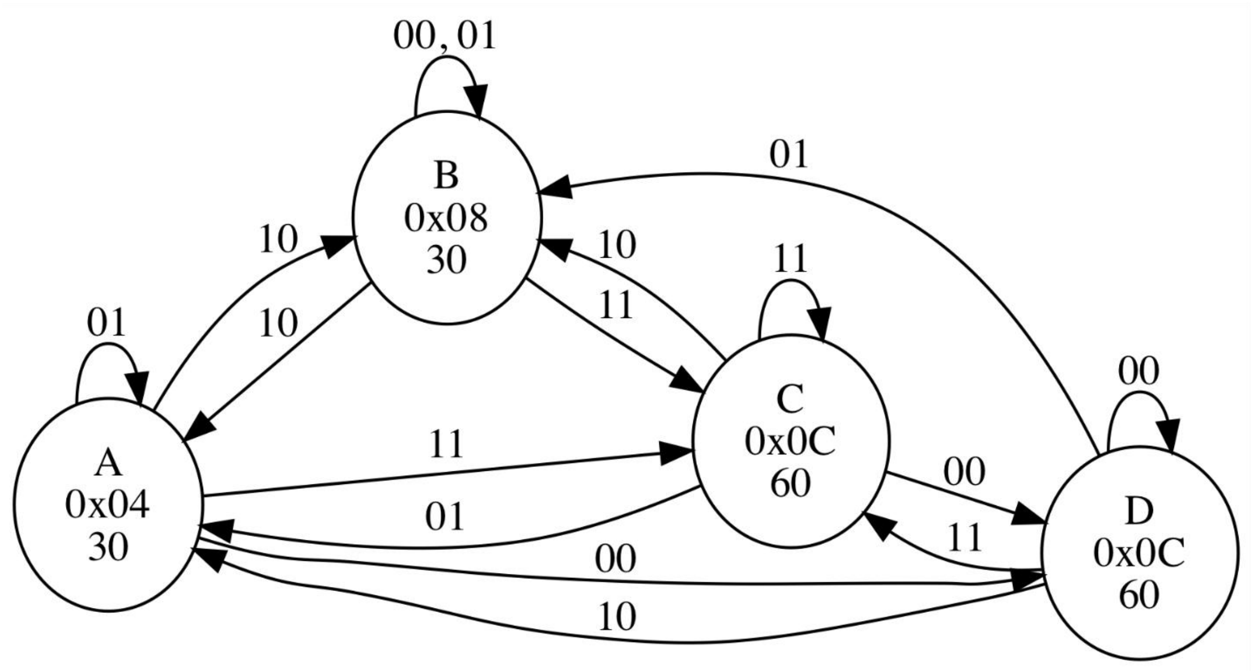
```
void Fun1(void){
uint16_t i = 1;
while (i<=100)
Output(i);
i = i+1;


}
```

11. What is the value of Diff when the following function is executed?

```
uint8_t Data[4] = {1,2,4,8};
uint8_t Diff;
void Fun3(void){
Diff = Data[1] - Data[2];
```

12. Explain difference between Moore and Mealy machine?

13. What is the output of state B? How much delay is there for state D?



14. Provide the state table for diagram above?

15. When the SysTick counter goes from 1 to 0, the Count flag in the NVIC_ST_CTRL_R register is set, triggering an interrupt.

16. The structure and the main program is fixed. Show the C code for FSM definition, and specify the initial state.