EE690

Embedded Systems Design
Lab Assignment 9
EK-TM4C123: Programming from Scratch

Discussion of Vector Table:

The allocation of memory in Tiva EK-TM4C123 was discussed, and the vector table of the ARM Cortex M4 was looked at specifically. It is seen that in the vector table,

- Position 0 is reserved for the Stack Pointer
- Position 1 is the initial Program Counter
- Position 2 to 15 are Core Exceptions
- Position 16 onwards are microcontroller specific interrupts

Then, the various exceptions and faults which can occur in the tm4c123gh6pm microcontroller are discussed. The core vector table components are:

- The RESET Exception
- Different faults were also discussed.
    - Bus error: Error caused when trying to access an memory location
    - Memory Usage error: Error caused by misusing data in a location, such as trying to execute a data location as if it was a program line.
    - Memory Protection related faults
- SVCall and PendSV, used for RTOS implementation
- Systick
- Non Maskable Interrupts

Writing a Minimal Program:

Here, the startup.c code is rewritten, to accommodate only the bare necessities required to run a program on the tiva microcontroller. The startup code is rewritten as follows:

- Firstly, the vector table is defined, which contains 32 bit entries.
    - The first entry is the end of the Stack Pointer
    - Then, the Reset_Handler is defined
    - The rest of the 14 core exceptions are populated with the Default_Handler
    - The tiva-specific exceptions are ignored.
- The Default_Handler and Reset_Handler are defined.
- The Reset_Handler is defined to:
    - Perform C Runtime Initialisation
    - Perform System Initialisation

    - Call main()
- C Runtime Initialisation and System Initialisation are also defined
- A program is written to toggle an LED on the TMC123GH6PM, using the modified startup file
    - The tmc123gh6pm.h header file is not used, hence the memory locations are manually defined
    - The code is written to toggle on the green LED on the tiva board, and is executed successfully.

Discussion on C Programming:

- C program is made of statements, which are preprocessor and C statements.
- C statements themselves are of 2 types; Declaration and Definitions.
    - When a C compiler compiles declarations, no code is generated, and hence no memory is used. On compilation of definitions, code is generated and memory is used as definitions contain instructions.
    - Definitions are of 2 types; Function definitions and Data Definitions.
        - Function definitions comprise of expressions
        - Data definitions can include temporary/local data, static data; which can be read only, initialised and uninitialised variables; and lastly, dynamic data.
- Preprocessing of main.c is carried out in the terminal, and the preprocessed file main.i is generated.
- The file is then compiled (but not assembled yet), to generate a main.s file.
- The .s file is then converted to binary in the next stage and converted into an object file (.o file) by the assembler (also called assembly).
- Linker file converts .ld file into an output map file. The final generated file is tia.out
- The tiva.out is flashed onto the tiva board and run.
- Proper functioning of the program is verified by observing toggling on of green LED.