

COMPUTER ORGANIZATION (ELE408) LAB 4

Interrupt Processing and DMA Operations of the TWR-K70F120M

1. Pre-lab Report

- Y/ Read Chapters 21 (DMAMUX), Chapter 22 (eDMA), Chapter 45 (PIT) and Chapter 55 (SPI) of TWR-K70F120M Reference Manual.
- Y/ Read this lab handout carefully. Take notes and prepare the programs required in this handout.
- Y/ Review the class notes regarding exception processing and DMA operations.

2. Objectives of this Lab

The purpose of this lab is to learn and gain first hand experiences on interrupt processing and DMA operations of the ARM Cortex M4 processor. You will apply your knowledge and basic concepts of exception processing; interrupt handling, and DMA operations that you have learnt in the lectures to the lab experiments.

Specifically, you will learn and exercise the basics of

- 2.1. Exception priorities, vector tables, and handler routines;
- 2.2. How to configure and program the interrupt controller to process various interrupt events and functions;
- 2.3. How to transfer large block of data without direct involvement of CPU;
- 2.4. How to save power by disabling modules that are not in use.

3. Basics

As discussed in the classes, interrupt is one of the most important mechanisms used in a computer system to handle various unexpected events such as bus errors, divide by zero, access errors, and so forth. It is also the most important interfacing technique to allow smooth communication between high speed devices such as CPU and RAM and low speed devices such as disks, displays, serial ports, printers etc. To design a microprocessor based system for any application, interrupt is one of the key mechanisms an engineer should master.

In many computer applications, it is necessary to move large amount of data between RAM and disk, between RAM and network, or between RAM and other peripheral devices. DMA (Direct Memory Access) is a technique allowing large block of data to be transferred between devices without direct involvement of CPU. It is therefore important to be able to configure and program DMA operations in a computer system.

The ARM Cortex M4 processor has a hardware interrupt controller and an enhanced 32-channel DMA controller. In this lab, you will configure and program these two controllers to implement various block data movement operations using DMA, and to handle error conditions, reporting DMA operation progress, signaling completion using the interrupt controller.

4. Experiment Procedures

This lab consists of three parts: 1) experiments on DMA operations, and 2) experiments on interrupt handling, 3) setting various power saving modes and observe power savings.

Main Functions To Be Implemented: In this lab, you will carry out data transfers between a RAMDisk and the system RAM using DMA controller. You are going to deal with three large matrices stored in the RAMdisk in Row-Major. Load the first two matrices into the system RAM block 1 and store them in Row-Major. Load the third matrix into RAM block 2 and store it in column-major. These two load operations should be performed on two different channels and optionally linked. Perform the matrix addition of the first matrix and third matrix. Store the sum matrix in RAM Block 3. Use DMA again to store the result matrix in RAM block 3 to RAMdisk.

You should be able to report transfer progress by sending a message to a programmer whenever the transfer is complete. You should also be able to handle error conditions such as configuration error by sending error messages to the programmer.

4.1. DMA Configurations and Moving Data Blocks

Preparation:

- Design a program to configure the DMA:
 - Select two DMA channels you would like to use
 - Enable Error Interrupt so that you can handle different error conditions
 - Enable E_LINK
 - Enable INT_HALF and INT_MAJOR
 - Use nested data transfers by configuring minor loop and major loop
- Design a program to carry out matrix addition
- Allocate RAMDisk, RAM Block 1, Block 2, and Block 3, respectively.
- Design a program to initialize RAMDisk

In the Lab:

- Configure the DMA using your program
- Initialize RAMDisk
- Transfer data as specified in the functional description
- Carry out the matrix addition as soon as data transfer of the two matrices is done
- Check transfer results and verify correctness

4.2. Interrupt Handling

Preparation:

- Write interrupt handlers for possible error events in DMA processing
- Write interrupt handlers to report data transfer progress

In the Lab:

- Run your program to configure the interrupt controller
- Store your interrupt handlers in the system RAM and update corresponding vector entries in the interrupt vector table
- You should have at least two interrupt handlers working properly: Configuration error handler, and completion message triggering matrix addition
- Artificially generate error conditions by writing different values to TCDn_CITER and TCDn_BITER
- Record your results and show them to the TA or the instructor

4.3. Power Saving Modes

Preparation:

- Design an assembly program to configure the power management module with different modes of operations: run, wait, doze, and stop on different peripheral devices

In the Lab:

- Configure your computer to different power saving modes and run programs for each mode
- Measure power consumption for each case
- Measure execution time of each run and compare them
- Discuss trade-off with your partner and the TA

5. Lab Report Requirements

In your lab report, you should discuss your designs, trade-offs between performance and power, Explanation and interpretation of your results are very important. The lab report will be graded based on your report and discussions. Total mark for the report is 100 points.

- Prelab report: 20 points
- Successful experiment: 50 points
- Results analysis, interpretation, and discussions on your design and engineering constraints: 30 points

In the following items, numbers inside each bracket $i^{\circ}[]i\pm$ indicates the point you will earn on a satisfactory report and discussions.

In your discussion and explanations of your results, you should consider the following:

- A.** What knowledge of mathematics, science and engineering have you applied in this lab? [5]
- B.** How did you design the lab (including architecture, flowchart, programs, etc.)? How did you conduct the experiment? What is your interpretation and conclusion on the experiments? [5]
- C.** Who is your team partner? How did you collaborate with each other and what roles did each of the team members play in the lab? [5]
- D.** What specific modern engineering tools have you used in your experiments. What specific techniques and skills have you learnt from the lab? [5]

Did your design consider the following constraints? If yes, how did you make your design decisions?

- Economic Constraint (performance/cost ratios) [3]
- Manufacturability, Modularity and Expandability Constraints [3]
- Environmental Constraint (power consumption) [2]
- Sustainability: Is your design and implementation sustainable? [2]

For each of the above programs hand in the debugged source code with comments; the machine code is not necessary. Note that both hardcopy and a file with programs are required for the report. Be very specific with your comments that explain what you are doing and why you are doing it.