EE 675: Microprocessor Applications in Power Electronics - Assignment 3

Jayesh Choudhary 170070038

1 Question 1

Write a C program to print data from the blocks of a TMS320F28379D bootstrapable .bin binary file.

For this question, a simple C program is given named 'rbin_170070038.c'. To read the binary data fread() and fseek() functions are used. The invocation for this program is given as **./rbin impy.bin** where the argument is the .bin file that we want to read and if 1 argument is not given or file is not found, an error message is displayed saying "File not found".

1.1 Algorithm

- First we check if the invocation is correct or not i.e. if only 1 arguement is there and if it could be opened or not.
- Then we read the first 2 bytes which tell us about the width of the instruction. If the bytes are **0x08aa**, then the width is 8 bits otherwise 16 bits. Typically it is 8 bits for our microprocessor.
- Then we move to the 23rd byte or byte[22] to read the block-size of the block that follows.
- Block information starts from 23rd byte. First 2 bytes tell us the block-size. If it is 0x0000, then it means we have reached the end of the file. We use this as the stopping condition for the while loop.
- Next 4 bytes store the address of the first block entry.
- From next byte on, data entries start. We then print these data entries in a for loop using printf() statement and using the iterator to get the address of the block entries.
- Once we read 0x000 as the block size, we break the while loop and close the file.

2 Question 2

Write a C program to read upto 8 words from the TMS320F28379D processor using the firmware of the ccm interface

For this question, a simple C program is given named 'rcom_170070038.c'. To open and close the ports, open_port() and close_port() of the sportf.c file were used. For reading and writing from the SPI port, read() and write() functions were used. The invocation for this program is given as **./rcom 1 8a00 6** where the argument 1 represents the port being used ('/dev/ttyUSB1'), the second argument 8a00 represents the address from where we have to read the data and argument 6 represents how many words are being read.

2.1 Algorithm

- First we check if the invocation is correct or not and if the port could be opened or not.
- We concatenate the first argument with the string "/dev/ttyUSB" to get the portNumber which will be used as argument to open the port. We then open the port with the baud rate specified as 115200
- We take the address as the second argument and left pad it with zeros to get 8 byte character array memoryAddress.
- We then initialize the 14 byte command header with first 2 bytes as 0x00, next 4 bytes with the address and the 7th byte or byte[6] with arguement 3 i.e. read size or how many words are being read. Here each character of the memoryAddress represents 4 bits of the address and so we extract the byte information for the command header by concatenating 2 characters and then converting it to 1 byte char value which is then fed to the command header.
- After that, we write the command header in the port byte-wise using write() function
- After that we read the values byte-wise and store it in an array named recieved.
- Then we concatenate 2 bytes to get a word since a word is of 2 bytes and then print the value using printf() function.
- Finally we close the port with close_port() function