Experiment 3: Advanced Assembly Coding

16.10.2019

Res. Asst. Yusuf Huseyin Sahin
sahinyu@itu.edu.tr

"Yes. Well, Dr. Shuman tells me that in theory there is nothing the computer can do that the human mind cannot do. The computer merely takes a finite amount of data and performs a finite amount of operations on them. The human mind can duplicate the process."

The president considered that. He said, "If Shuman says this, I am inclined to believe him - in theory. But, in practice, how can anyone know how a computer works?"

Brant laughed genially. "Well, Mr. President, I asked the same question. It seems that at one time computers were designed directly by human beings. Those were simple computers, of course, this being before the time of the rational use of computers to design more advanced computers had been established."

...

Nine times seven, thought Shuman with deep satisfaction, is sixty-three, and I don't need a computer to tell me so. The computer is in my own head.

And it was amazing the feeling of power that gave him.

Isaac Asimov, The Feeling Of Power

1 Introduction and Preliminary

In this experiment we will obtain more experience on the MSP430 board and assembly coding. We will first find an appropriate way to obtain modulus operation using assembly language. Then, we will find the first 50 prime numbers. Lastly, we will show Goldbach's conjecture using these numbers. Before coming to the lab, it is strongly suggested to study on the given algorithm.

2 Part 1: Modulus Operation

For modulus operation we will perform some steps from Russian Peasant Division which is very useful for binary systems. RPD consists of the following steps for dividend A and divisor B:

- Create variables C and D, initialize them with B and A's values respectively.

- While C is not greater than A/2, multiply it by 2.
- While B is not greater than D:
  - Subtract C from D wherever D is greater than or equal to C.
  - Divide C by 2.
- The final value in D gives you the remainder. If you subtract this number from A, this new A can be represented by summation of previously used C values. After finding the proper C values, you can find their appearance iteration in the first loop, raise them to 2nd power, and sum them.

An example implementation is given in the table below.

A	B	C	D	Explanation
151	8	8	151	-
151	8	**16**	151	Multiplying C by 2
151	8	**32**	151	-
151	8	**64**	151	-
151	8	**128**	151	Now it's greater than A/2
151	8	**64**	**23**	151-128=23
151	8	**32**	23	-
151	8	**16**	23	-
151	8	**8**	**7**	23-16=7
144	8	8	7	151-7= 144
128+16	8	8	7	144=128+16
				128=C[4] 16=C[1]
				$2^4 + 2^1 = 18$

Write the assembly code which calculates the modulus.

3 Part 2: Prime Numbers

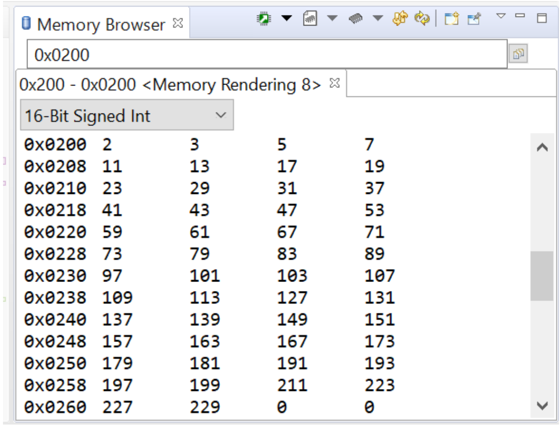
Write the assembly code to find the first 50 prime numbers and save them to a memory address. To allocate space in memory, you can use ".space" directive.

```
                .data
primes          .space 100
array1          .space 100
array2          .space 100
```

Here, 100 bytes of memory are allocated to keep prime numbers. The other two arrays will be useful for the last part of the experiment. Inside the code part, you can use *#primes* to reach the address of the first element.

```
    mov.w #primes, R10; First address for prime numbers is assigned to R10
    ...
    mov.w   R5, 0(R10); Some value is copied to that address.
```

The example output can be seen in the figure below.



4 Part 3: Goldbach's Conjecture

Goldbach's Conjecture is a still unproven conjecture stating "Every even number greater than 2 is the sum of two primes."[1]

Write the assembly code to find prime numbers which can be summed up to obtain the even numbers between [200-298], save them to the memory addresses shown by #*array1* and #*array2*.

[1]Doxiadis, Apostolos. Uncle Petros and Goldbach's Conjecture: A Novel of Mathematical Obsession. Bloomsbury Publishing USA, 2001.