Experiment 8: Random Number Generation

21.11-23.11.2022
Res. Asst. Altay Ünal
unal21@itu.edu.tr

Alea iacta est.

Julius Caesar

1 Introduction

In the following experiment, you are going to implement a sequence generator and a random number generator. Note that this is the last experiment and we are expecting a better coding such as using subroutines with operands transferred using stack. You will not get much of help for the syntax in this experiment.

2 Part 1

In this part, you are expected to generate a sequence using "Blum-Blum-Shub" approach. Please follow the given "Blum-Blum-Shub" algorithm and implement it as an interrupt subroutine. Use P2.5 as button to interrupt the microcomputer and show the numbers of the sequence on the 7-segment display.

```
p = 11, w = 13, s = any seed other than 1 and 0
M = p * q;
r = power(s,2) mod M;
s = r;
```

You can use 5 as your initial seed.

3 Part 2

In this part, you should implement a random number generator using "Middle Square Weyl Sequence" approach. Please check the following Middle Square Weyl Sequence algorithm and implement it as an interrupt subroutine. Use P2.5 as button to interrupt the microcomputer to generate a new random number between 0-128 and show it on the 7-segment display.

```
x = 0, w = 0, s = any seed
x = square(x);
x = x + (w = w + s);
r = (x>>4) | (x<<4);
```

4 Part 3

In this part, you are going to use the Random Number Generator in Part1. Use P2.5 as button to interrupt the microcomputer to generate 128 random number between 0-8(8 is not included) and write it into preallocated memory locations. Check that how much the distribution of the random numbers generated from your random number generator close uniform. You can use counting to check the uniformity. Show your work on the memory using preallocated memory locations. Show your work with different amount of random numbers and with different intervals of random numbers.