

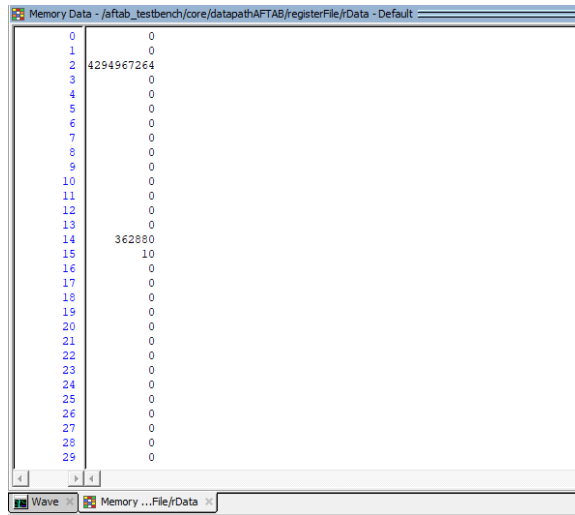
In the name of God

Core-Based Embedded System Design

Mohammad Yahyapour
CA1

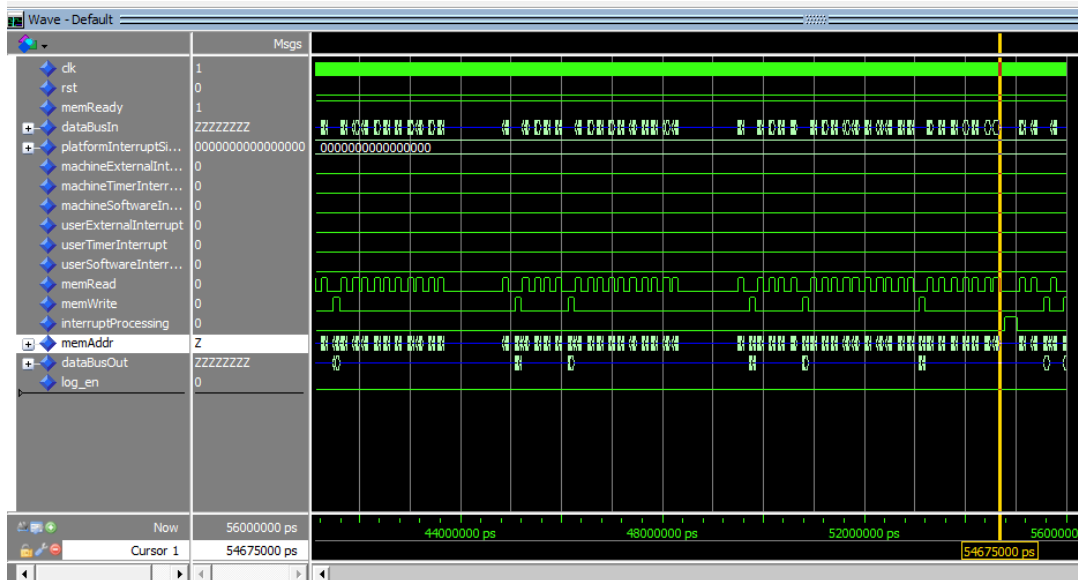
Part A

1)



Register	Value
0	0
1	0
2	4294967264
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	362880
15	10
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0

As we can see from the above picture, a factorial of 10 is calculated and stored in 15th register.



Finishing doing all instruction in 54675 ns.

One iteration of while took about 3450 ns.

2)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

The calculation that the processor is supposed to handle it.

Register	Value
1	0
2	4294967248
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	1048640
13	6
14	22
15	2
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0

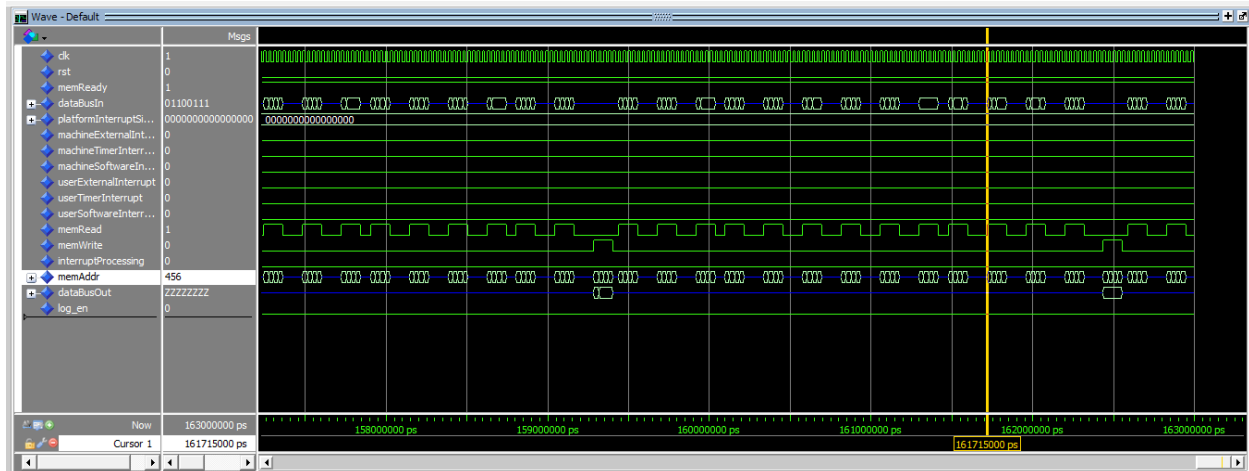
The intermediate calculation of C12

Register	Value
1	0
2	4294967248
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	1048640
13	15
14	43
15	1
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0

The intermediate calculation of C21

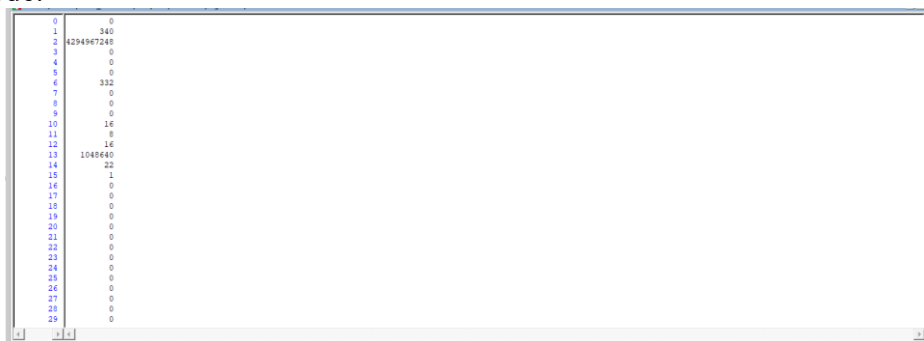
Register	Value
1	0
2	4294967248
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	1048640
13	15
14	50
15	1048652
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0

The intermediate calculation of C22



The instructions almost finished at 161715 ns.

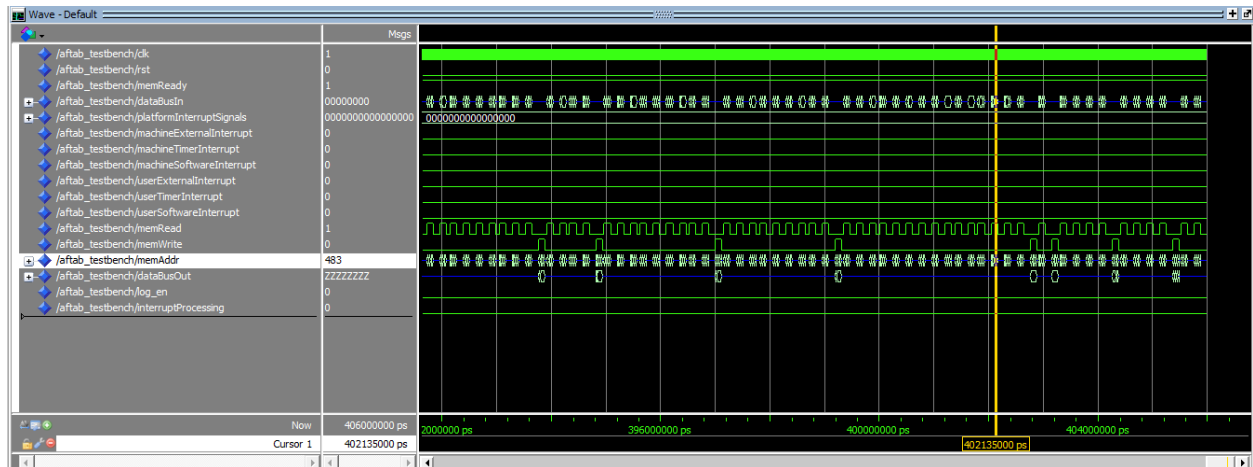
The other part of this section needed to complete this part without using the multiplication instruction and instead achieve it through repeated addition in a loop. Due to some issues the assembly code of mul function was transferred to the end of the code and then transformed to machine code.



The intermediate calculation of C12



The intermediate calculation of C22



The last instruction which is ra and located in address of 480 of memory done at 402135 ns.

Comparing Execution Times

$$\frac{402135}{161715} = 2.48$$

Using the multiplication instruction, we achieved a $24.8\times$ speedup compared to repeated addition. Due to the fact that calling functions have an overhead and multiplication with addition is at least 26 ($5+6+7+8 = 26$) time slower than the other one. The reason for this can be better understood by analyzing the C++ implementation of the mul function. The speedup value largely depends on the sum of the elements in the B matrix.