Exercise 1:

```
PUSH M2
PUSH M4
MUL                 //M2*M4
POP M5
PUSH M5
PUSH M3
ADD                 //M3+(M2*M4)
POP M6
PUSH M1
PUSH M2
ADD                 //M1+M2
POP M7
PUSH M6
PUSH M7
MUL
POP M2              //(M1+M2)*(M3+(M2*M4))
```

Exercise 2:

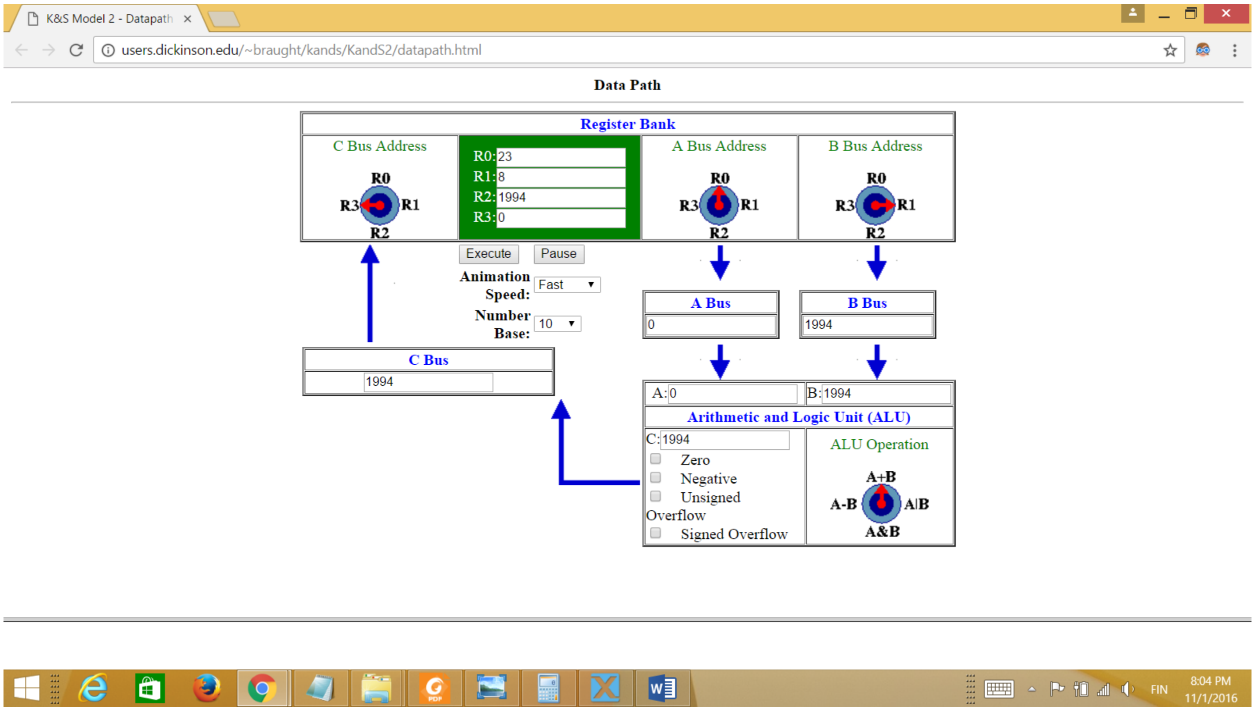
```
__attribute__(( naked )) void asm_test(int *a, int *b, int *c, int *d)
{

asm volatile
(
            "push {r4, r5, r6, r7} \n"
            "ldr r4, [r0] \n"
            "ldr r5, [r1] \n"              //r5 = M2
            "adds r6, r4, r5 \n"           //r6 = M1+M2
            "ldr r4, [r3] \n"              //r4 <- M4
            "muls r4, r5, r4 \n"           //r4 = M2*M4
            "ldr r5, [r2] \n"              //r5 <- M3
            "adds r7, r4, r5 \n"           //r7 = M3+(M2*M4)
            "muls r7, r6, r7 \n"           //r7 =  (M1+M2)*(M3+(M2*M4))
            "str r7,[r1] \n"
            "pop {r4, r5, r6, r7} \n"
            "bx lr \n"
);
}
```

```
void fail() {
       printf("Failed\n"); // set a break point here
}
```

Exercise 3:

Part A:



users.dickinson.edu/~braught/kands/KandS2/datapath.html

Data Path

Register Bank

R0: 23
R1: 8
R2: 1994
R3: 0

Animation Speed: Fast

Number Base: 10

A Bus: 0

B Bus: 1994

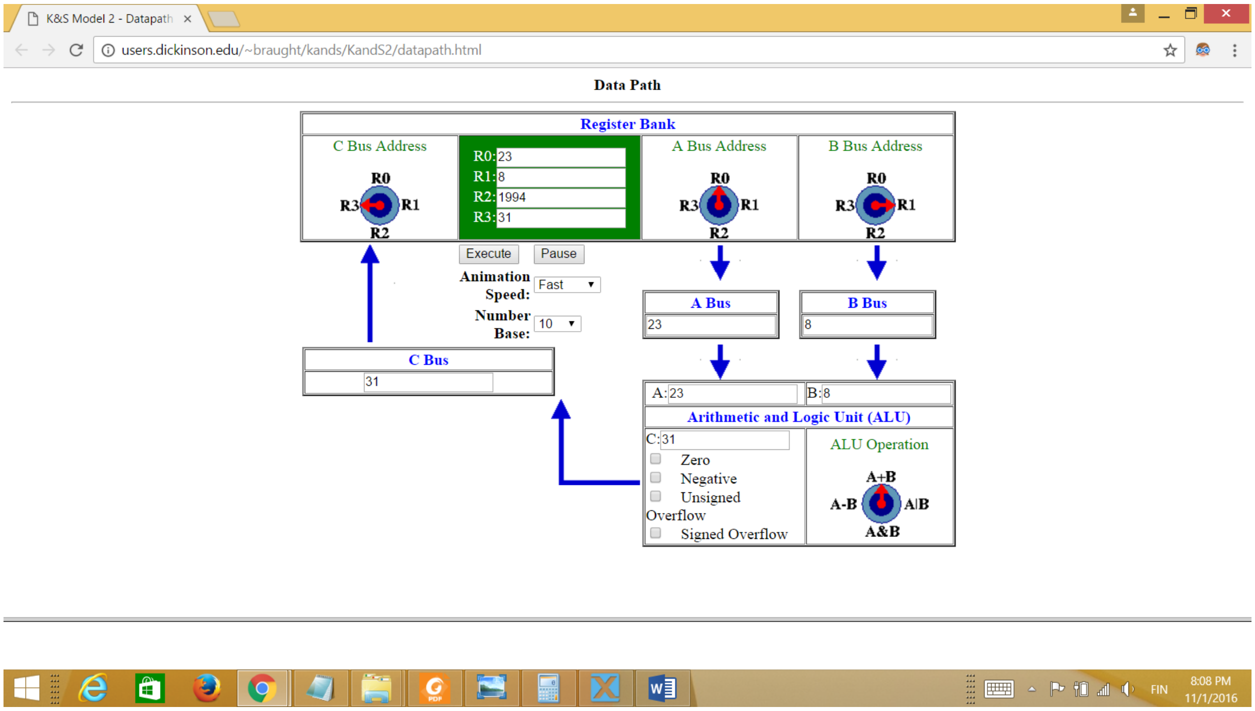
C Bus: 1994

A: 0
B: 1994

Arithmetic and Logic Unit (ALU)

C: 1994
□ Zero
□ Negative
□ Unsigned Overflow
□ Signed Overflow

8:04 PM
11/1/2016



users.dickinson.edu/~braught/kands/KandS2/datapath.html

Data Path

Register Bank

R0: 23
R1: 8
R2: 1994
R3: 31

Animation Speed: Fast

Number Base: 10

A Bus: 23

B Bus: 8

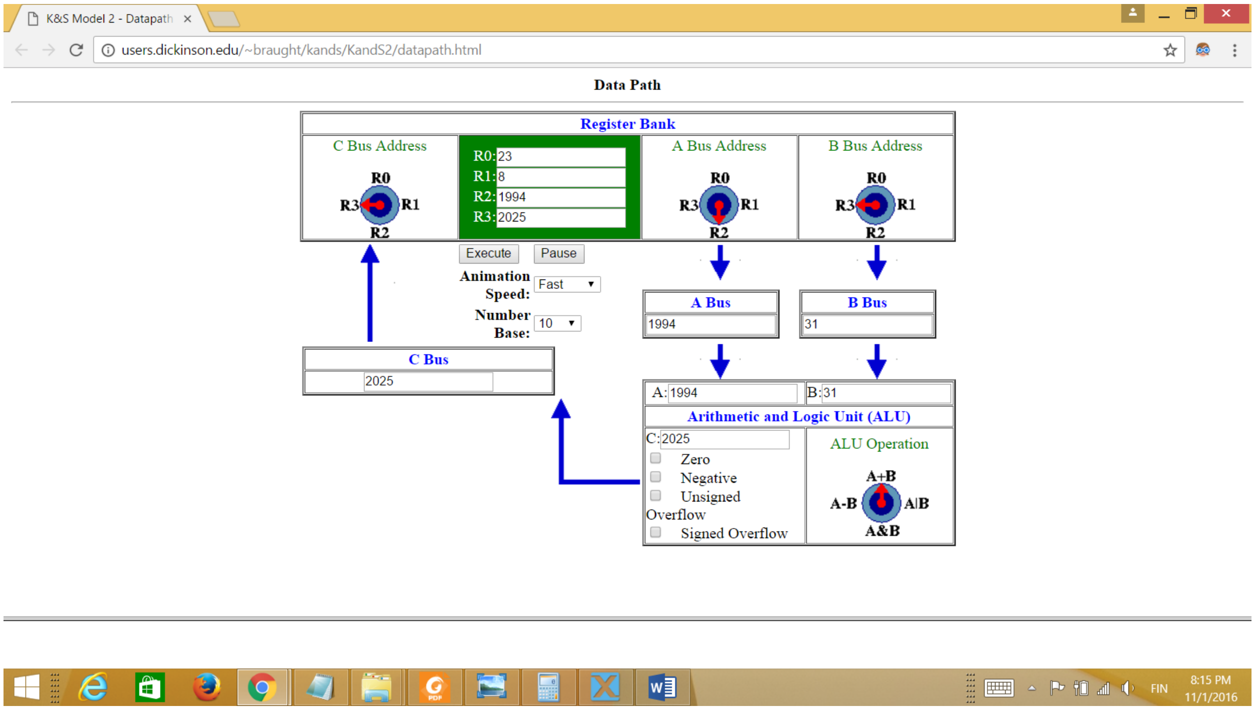
C Bus: 31

A: 23
B: 8

Arithmetic and Logic Unit (ALU)

C: 31
□ Zero
□ Negative
□ Unsigned Overflow
□ Signed Overflow

8:08 PM
11/1/2016



Data Path

Register Bank

Register	Value
R0:	23
R1:	8
R2:	1994
R3:	2025

C Bus Address: R3 · A Bus Address: R2 · B Bus Address: R3

Animation Speed: Fast

Number Base: 10

A Bus: 1994

B Bus: 31

C Bus: 2025

Arithmetic and Logic Unit (ALU)

A: 1994

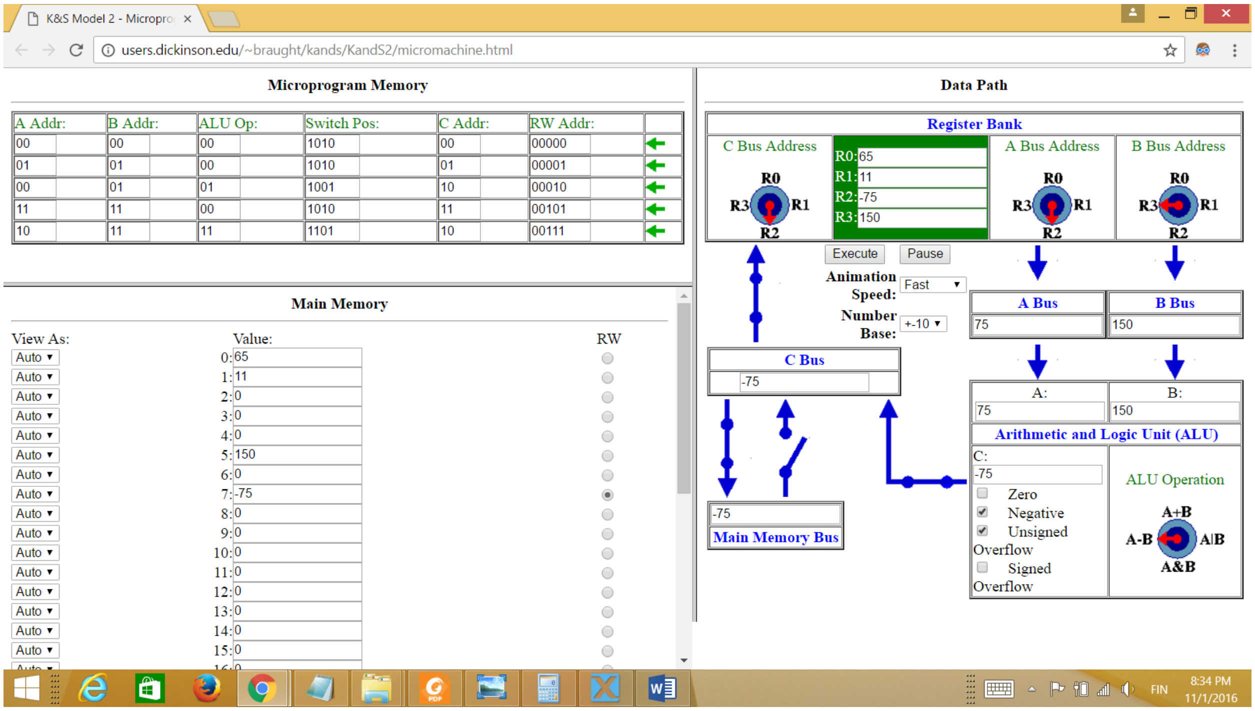
B: 31

C: 2025

□ Zero
□ Negative
□ Unsigned Overflow
□ Signed Overflow

ALU Operation: A+B

Part B:



Microprogram Memory

A Addr:	B Addr:	ALU Op:	Switch Pos:	C Addr:	RW Addr:
00	00	00	1010	00	00000
01	01	00	1010	01	00001
00	01	01	1001	10	00010
11	11	00	1010	11	00101
10	11	11	1101	10	00111

Main Memory

View As:	Value:	RW
Auto	0: 65	○
Auto	1: 11	○
Auto	2: 0	○
Auto	3: 0	○
Auto	4: 0	○
Auto	5: 150	○
Auto	6: 0	○
Auto	7: -75	●
Auto	8: 0	○
Auto	9: 0	○
Auto	10: 0	○
Auto	11: 0	○
Auto	12: 0	○
Auto	13: 0	○
Auto	14: 0	○
Auto	15: 0	○

Data Path

Register Bank

Register	Value
R0:	65
R1:	11
R2:	-75
R3:	150

C Bus Address: R2 · A Bus Address: R2 · B Bus Address: R3

Animation Speed: Fast

Number Base: +-10

A Bus: 75

B Bus: 150

C Bus: -75

Main Memory Bus: -75

Arithmetic and Logic Unit (ALU)

A: 75

B: 150

C: -75

□ Zero
☑ Negative
☑ Unsigned Overflow
□ Signed Overflow

ALU Operation: A-B