Benjamin Linam

Bml42@students.uwf.edu

EEL 4744L: Microprocessor Applications Laboratory

Lab 6: Logic State Analysis of the MC68HC11 Bus

3/21/2018

Objective

The object of this lab is to understand the timing of bus cycles and the operation of the MC68HC11 in expanded mode.

Introduction/Background/Theory

To make the HC11 more efficient, its developers created a way to reduce the number of pins required for reading and writing to memory. The processor has sixteen pins that are used to read and write from AD15-AD8 and AD7-AD0. During the first half of an E-clock cycle, the processor is required to access an address in memory and the AS (address strobe) signal drives the lower order pins (A7-A0). After the first half of the cycle, those same low order pins (D7-D0), are used to read and write data.

Being able to multiplex those pins for use as either an address bus or data bus cuts down on the overhead of having more pins to do the same job. This lab will show the relation of signals moving across pins such as AD15-AD8, AD7-AD0, the E-clock, AS, and R/W* and how they all work together to execute a set of instructions.

Procedure

1. The lab required downloading the following ASM code, shown in Fig 1., to the HC11. A list file was also to be created showing the steps performed by the processor at each cycle (Fig 2.).

ASM Code for lab 6

```
HPRIO   EQU $103C
MASK    EQU $F5

        ORG $0200
M       RMB 1       ; RESERVE 1 BYTE FOR M AT BEGINNING OF EXTERNAL RAM
N       RMB 1       ; RESERVE 1 BYTE FOR N AT NEXT MEMORY LOCATION
SUM     RMB 1       ; RESERVE 1 BYTE FOR SUM AT NEXT MEMORY LOCATION

        ORG $B600
        LDAA #MASK  ; CHANGE HPRIO FOR INTERNAL READ VISIBILITY
        STAA HPRIO
        LDAA #5     ; INITIALIZE M TO 5
        STAA M
        LDAA #4     ; INITIALIZE N TO 4
        STAA N
LOOP    LDAA M      ; LOAD M INTO ACCA
        LDAB N      ; LOAB N INTO ACCB
        ABA         ; ACCA = ACCA + ACCB
        STAA SUM    ; STORE SUM
        BRA LOOP    ; LOOP FOREVER
        SWI
```

Figure 1: Assembly language code showcasing the infinite loop required for analysis

```
                    HPRIO   EQU     $103C
                    MASK    EQU     $F5

                            ORG     $0200
                    M       RMB     1       ;Reserve 1 byte for M at beginning of
                    external RAM
                    N       RMB     1       ;Reserve 1 byte for N at next memory
                    location
                    SUM     RMB     1       ;Reserve 1 byte for SUM at next
                    memory location

                            ORG     $B600
B600 86 F5                  LDAA    #MASK   ;Change HPRIO for internal read
                    visibility
B602 B7 10 3C               STAA    HPRIO
B605 86 05                  LDAA    #5      ;initialize M to 5
B607 B7 02 00               STAA    M
B60A 86 04                  LDAA    #4      ;initialize N to 4
B60C B7 02 01               STAA    N
B60F B6 02 00       LOOP    LDAA    M       ;load M into [A]
B612 F6 02 01               LDAB    N       ;load N into [B]
B615 1B                     ABA             ;[A] = [A] + [B]
B616 B7 02 02               STAA    SUM     ;store SUM
B619 20 F4                  BRA     LOOP    ;Loop forever
B61B 3F                     SWI

Symbol Table

MASK                00F5
M                   0200
N                   0201
SUM                 0202
HPRIO               103C
LOOP                B60F
```

Figure 2: List file created after compiling the above ASM code.

2. During the lab, a logic analyzer was provided to connect to the HC11 and produce a time chart illustrating the operations occurring at pins AD15-A8 and AD7-AD0, as well as the E-clock, R/W*, and AS pins. Below, Fig3. shows the values of each of the signals during a single loop of the branching instructions shown in Fig. 1.

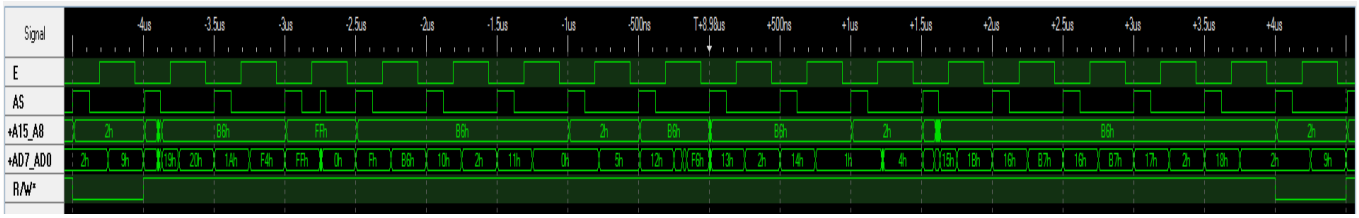

Figure 3: A screen capture of the logic analyzer showing the values of the requested pins during a single loop

Conclusions

This lab was not inherently difficult, as it was mostly analysis of data. It was interesting to see the individual steps the HC11 takes to complete an assigned job.

Questions

1. Cycle-by-cycle execution table

Cycle	Address	Data	R/W*
1	B600	86	1
2	B601	F5	1
3	B602	B7	1
4	B603	10	1
5	B604	3C	0
6	B605	86	1
7	B606	05	1

8	B607	B7	1
9	B608	02	1
10	B609	00	0
11	B60A	86	1
12	B60B	04	1
13	B60C	B7	1
14	B60D	02	1
15	B60E	01	0
16	B60F	B6	1
17	B610	02	1
18	B611	00	1
19	B612	F6	1
20	B613	02	1
21	B614	01	1
22	B615	1B	1
23	B616	B7	1
24	B617	02	1
25	B618	02	0
26	B619	20	1
27	B61A	F4	1
28	B61B	3F*	1

*SWI is listed as taking 14 cycles to complete; however, the data bus is not essential after the first cycle.

2. The IRV bit is primarily used in expanded or special test modes as it is used in the debugging process of using a logic analyzer. It is tied to the E clock and is low when SMOD is low but high when SMOD is high.

3. During a cycle, when AS is 1, the AD7-AD0 bus lines start with an address and then alternate to data when AS is 0. For example, when the PC reaches address B619 and AS is 1, AD15-AD8 reads B6 while AD7-AD0 reads 19 followed by 20 when AS is 0.

4. The HC11 conducts a read cycle when the signal at R/W* is 1 and conducts a write cycle when the signal at R/W* is 0.