EE 382N-4 Embedded System Architecture
HW #2
Assigned: Feb 11th, 2011 Due: March 6th (ICS Program)
Assigned: Feb 14th, 2011 Due: March 7th, 2011 (Regular Program)

Compile the following FIR filter code for the ARM processor. Determine where the performance bottlenecks are with the C code by analyzing the resulting assembly language code. The issues to look at include register usage vs. memory/stack usage, loops and loop counters, conditional execution.

/* FILENAME = fir.c */

```
int main() {
}

/* FIR filter in direct form */

double fir(M, h, w, x)                  /* Usage: y = fir(M, h, w, x); */
double *h, *w, x;                       /* \(h\) = filter, \(w\) = state, \(x\)
= input sample */
int M;                                  /* \(M\) = filter order */
{
       int i;
       double y;                        /* output sample */

       w[0] = x;                        /* read current input sample \(x\) */

       for (y=0, i=0; i<=M; i++)
              y += h[i] * w[i];         /* compute current output sample \(y\)
*/

       for (i=M; i>=1; i--)             /* update states for next call */
              w[i] = w[i-1];            /* done in reverse order */

       return y;
}
```

The following makefile can be used to compile the code. The resulting assembly language will be in a file named fir.asm. Compile the code on the Linux machines in the LRC (soc1, soc2, soc3)

```
PROC=arm
TYPE=linux
VERSION=arm-2008q1
LIBPATH=/usr/local/$(PROC)/$(VERSION)/lib
INCPATH=/usr/local/$(PROC)/$(VERSION)/include
CC=$(PROC)-$(TYPE)-gcc
NM=$(PROC)-$(TYPE)-nm
CFLAGS= -g -I$(INCPATH) -L$(LIBPATH)
STRIP=$(PROC)-$(TYPE)-strip
OD=$(PROC)-$(TYPE)-objdump
ODFLAGS=-Sd

fir:
             $(CC) fir.c $(CFLAGS) -o $@ ;\
             $(NM) --debug-syms --numeric-sort $@ > fir.gysm
             $(OD) $(ODFLAGS) fir > fir.asm

clean:
             rm fir fir.asm
```

EE 382N-4 Embedded System Architecture
HW #2
Assigned: Feb 11th, 2011 Due: March 6th (ICS Program)
Assigned: Feb 14th, 2011 Due: March 7th, 2011 (Regular Program)

There are at least 4 major bottlenecks in the FIR code. Highlight each one and describe a solution for the bottleneck.

The report must be in the following form for each bottleneck

C-Code	Assembler Code	Performance Bottleneck	Solution
Highlight the C-Code which is causing the performance problems	Carve out the Assembly language that corresponds to the C-Code	Describe what is causing the performance problem.	What is the solution?