

ARRL Handbook CD

Template File

Title: DSP-3 Filter

Chapter: 16

Topic: A Digital Signal Processor

Template contains:

Additional text.

PC board etching and solder-mask patterns.

Component layout diagram.

Template for W9GR DSP Filter

Writing Your Own DSP Firmware

Some might want to write their own firmware for the digital signal processor. To do this, it is first necessary to develop a familiarity with the algorithm or function to be implemented. This may entail quite a bit of study and mathematics, depending on the function. On the other hand, if a particular filter specification is the objective, there are several digital filter design programs and textbooks which can lead the programmer through a "cookbook" process, but some math will still probably be involved. The next step is to code the design into TMS320C1x assembly language. The third step is to *assemble* the program, using any of several commercially available TMS320C1x cross-assemblers. At this point it may be helpful to simulate the TMS320C1x program using a commercial simulator in non-real time, or to execute it in real time with real signals on an emulator, the TI "EVM" evaluation module, or one of the TI "DSK" DSP starter kits. Finally, the assembler output is converted to firmware by burning the program into EPROMs or a CPU with internal (E)PROM.

There are two ways to run your own DSP programs on this hardware platform. The first way is to program a TMS320P15 ("OTP" or one-time-programmable) or TMS320E15 (windowed erasable part, no longer manufactured by TI).

The other way to run your own programs is to make use of the provision for external 57C43 high speed EPROMs.

(Either approach requires a device programmer which is capable of burning the special high speed EPROMs or the TMS320 CPU chips that have internal program memory.)

The MC/MP- pin controls whether internal or external program memory is used. "MC" stands for "microcomputer" mode where internal program memory is used (logic 1), and "MP-" refers to "microprocessor" mode where the program firmware is in external memory (a logic 0). The standard TMS320C10 or TMS320C15 has no internal program memory, so only the "MP-" microprocessor mode is meaningful. The "MC" mode is used by the TMS320E15 and TMS320P15 programmable members of the TMS320 family. (The TMS320E15 contains internal program memory EPROM whereas the P15 part is the windowless one-time programmable version of the same. Unfortunately, the TMS320E15 is no longer being made by TI.)

To run programs from the external EPROMs, the MC/MP- jumper must be set to the MP- (logic zero) position, and a CPU of the TMS320C1x family must be installed which does not have the "security bit" programmed. (When the security bit of a TMS320P15 or TMS320E15 CPU is set, then the MC/MP- input to the chip is ignored, and the CPU will only be able to execute its internal program. So if you want to run your own DSP programs, you will not be able to use the preprogrammed CPU mentioned above.)

Another difference among members of the TMS320C1x family is internal data memory size. The TMS320C10 has 144 16 bit memory words while the TMS320C15, TMS320E15, and TMS320P15 all have 256.

The port addresses used in this digital signal processor are as follows:

INPUT PORT	OUTPUT PORT
PA0mode switch	DAC write
PA1A/D read	(not used)
PA2(not used)	LED bargraph
PA3(not used)	(not used)

Although IC2, the TCM320AC39 audio codec chip uses serial interfacing, the members of the TMS320C1x series of CPU chips used in this project do not have a serial port. Consequently, there is some "glue logic" to perform the serial to parallel and parallel to serial conversions and to provide the necessary clock and timing signals to IC2. To minimize the complexity of this glue logic circuitry, there are some constraints that DSP programmers must provide for.

First, the sampling rate is determined by software. Sampling rate is determined by how often the program writes samples to IC2. There is no built-in hardware sampling clock in the digital signal processor, so analog I/O will occur whenever an I/O instruction is executed. If the TMS320C1x code is written entirely "in-line" with a "IN" instruction at the top and an "OUT" instruction at the bottom of the main program loop, then input and output sampling rates will be constant. But if there is program branching, the various program execution paths will have to be timed and usually padded with NOP (no operation) instructions to make the sampling rate and sampling interval independent of the program branching.

Secondly, writing an output sample to IC2 not only updates the DAC, but it also initiates an A to D conversion. After at least 68 machine cycles, the input sample will be loaded into the shift register, ready to read at port PA1. In other words, an IN statement should not immediately follow an OUT statement - allow at least 68 machine cycles.

Third, IC2 is clocked from a 2.5 MHz clock produced by dividing the CPU's 5 MHz output by two. A consequence of this is that there must always be an even number of program steps between samples. (This can be easily adjusted, if necessary, by adding a NOP instruction.)

Audio input samples are 13 bits, two's complement, and are "left justified," meaning that they are read into the 13 most significant bits of the 16 bit data path of the TMS320C1x. The remaining 3 LSBs are set to zero by IC2.

Audio samples written to the DAC at IC2 must be rounded or truncated to 13 bits and should have their 3 LSBs set to zero. This can be accomplished with the AND instruction. Although it is a 13 bit DAC, the 3 LSBs are used to control the audio level at the DAC output. Setting the three LSBs to zero results in maximum audio gain. If you do not round or truncate your output words to 13 bits, you will have highly distorted output audio as IC2 switches its output level among 8 different levels from sample to sample!

The normal and maximum sampling rate allowed by IC2 is 5 MHz/650 or 7692 Hz. This corresponds to 650 machine cycles. Most TMS320C1x instructions take a single machine cycle, but some such as branches, I/O instructions, etc. take two or more. The number of machine cycles between analog I/O instructions should be counted and normally set to 650.

To drive the LED bargraph, write a word to output port PA2. A 0 in any bit position will turn on the corresponding LED segment, because the LEDs are connected between the latch output and the positive supply. An output word is 16 bits wide but there are only 10 segments in the LED

bargraph. The output word is "right justified," which means that the lower 10 bits of the word affect the bargraph, and the upper 6 bits are ignored. The LSB controls the leftmost LED segment.

The setting of the mode switch S3, is determined by reading port PA0. This is a 16 position binary switch encoded into 4 bits. The input word is "right justified," which means that the lower 4 bits of input the word contain the bargraph, and the upper 12 bits are undefined. Because the upper 12 bits are high impedance when reading the mode switch, their values cannot be relied upon to have any consistent value. Consequently, the upper 12 bits should be ignored by your software and/or forced to a known value with an AND or OR instruction.

Beginning at the 12 o'clock position and going clockwise viewed from the front of encoder switch S3, the 16 positions of S3 produce the following codes after the inversion produced by IC8:

1000 mode 1	0000 mode 9
1001 mode 2	0001 mode 10
1010 mode 3	0010 mode 11
1011 mode 4	0011 mode 12
1100 mode 5	0100 mode 13
1101 mode 6	0101 mode 14
1110 mode 7	0110 mode 15
1111 mode 8	0111 mode 16

A SPDT switch is connected to the BIO input on the CPU. This switch can be used to set options, etc., and its position can be read by using the BIO instruction.

Parts List

Unless otherwise specified, all resistors are 1/4 watt 5% and capacitors are 10% tolerance.

Any of the 74XX series TTL parts (marked with a * below) may be used in any the following variations: 74XX, 74LSXX, 74SXX, 74FXX, 74ALSXX, 74HCXX, 74HCTXX, 74AHCTXX, 74HCTLSXX, etc. For IC5, a HCMOS type is recommended (e. g. HCT, HCTLS, etc.) to minimize the effects of gate input current on the power on reset circuit. When mixing logic family types, bipolar devices (LS etc.) must always drive devices which accept bipolar input levels (e. g. HCT, HCTLS, etc.).

Quantity	Reference Designator(s)	Description
	C13, C14	22 pF silver mica or disk ceramic capacitor
	C7	0.001 uF ceramic 20%
2	C5, C8	0.01 uF ceramic 20%
12	C6, C10, C11, C15, C17, C18, C20, C21, C22, C23, C24, C25	0.1 uF monolithic ceramic, 20%
3	C2, C3, C4	1.0 uF electrolytic (16V or greater)
2	C1, C9	10 uF electrolytic (16V or greater)
3	C12, C16, C19	220 uF electrolytic (16V or greater)
1	R14	2.7 ohms
1	R16	10 ohms
1	R13	39 ohms
11	R8, R21, R22, R23, R24, R25, R26, R27 R28, R29, R30	330 ohms
1	R7	820 ohms
1	R15	1000 ohms
1	R2	1800 ohms
2	R10, R12	3000 ohms
2	R4, R5	6.2 K ohms
6	R1, R3, R17, R18, R19, R20	10 K ohms
2	R6, R9	100 K ohms
1	R11	100 K ohm pot
2	D1, D2	1N4001 silicon diode (diodes with higher voltage ratings such as 1N4002, 1N4003, 1N4004 etc. may also be used)
1	X1	20 MHz fundamental crystal
1	IC1	TMS320P15 DSP CPU programmed with firmware (available from Quantics, P. O. Box 2163, Nevada City,

		California 95959-2163 USA.)
1	IC2	TCM320AC39 audio codec (TCM320AC38
may be substituted)		
2	IC3, IC4	74299 shift register (*)
1	IC5	7400 quad NAND gate (HCMOS type
recommended) (*)		
3	IC6, IC7, IC11	7474 dual flip flop (*)
1	IC8	74368 hex inverter (*)
1	IC9	74139 decoder (*)
1	IC10	74374 octal flip flop (*)
1	IC12	LM380 audio power amplifier
(2)	IC13, IC14	WaferScale 57C43 EPROM
		(only used for user developed firmware)
1	VR1	7805 or LM340T-5 TO-220 voltage regulator
2	L1, L2	10 uH RF chokes
1	LED1	10 segment LED bargraph, Radio Shack
276-081B		
		or equivalent
1	J1	Coaxial DC power jack
2	J2, J3	1/8" audio jack
1	J4	1/8" stereo headphone jack
2	S1, S2	DPDT push-push switch
2		Switch caps for S1 & S2 (1 red; 1 gray)
1	S3	16 position binary rotary encoder switch
		(Noble SDB161 PH 20F-1-4-16-16PC-B)
1	S4	SPDT toggle switch (for BIO function)
1		TO220 heatsink
5		14 pin IC socket
2		16 pin IC socket
4		20 pin IC socket
1		40 pin IC socket
1		20 pin right angle socket (for LED bargraph)
2		knob
1		PC board
1		set of #4 hardware for VR1 & heatsink

including noise reduction adaptive filtering, automatic notch filtering, filtering for special modes including SSTV and several forms of FSK, narrow SSB filtering, CW filtering, a DTMF decoder, and a CTCSS decoder







