AMI

AMERICAN MICROSYSTEMS, INC.

S2814A

FAST FOURIER TRANSFORMER

Features

□ Based on AMI's Signal Processing Peripheral Chip (S2811)

□ Performs 32 Complex Point Forward or Inverse FFT in 1.3msec, Using Decimation in Frequency (DIF)

□ Transform Expandable either by Using Multiple S2814As (for Minimum Processing Time) or by a Single S2814A (for Minimum Hardware)

□ Operates with any 8 or 16 Bit Microprocessor, including S6800 and S9900. Optional DMA Controller Increases Speed

□ All Data I/O Carried Out on Microprocessor Data Bus

□ Basic Resolution of 57dB. Optional Conditional Array Scaling (CAS) Routine Increases Dynamic Range to 70dB

□ Optional Windowing Routine Incorporated to Permit Use of Arbitrary Weighting Function

□ Coefficient Generation On Chip, with Rotation Algorithm for Transform Expansion up to 512 Points

□ Optional Power Spectrum Computation

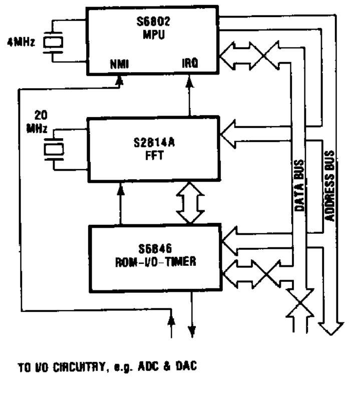
General Description

The AMI S2814A Fast Fourier Transformer is a preprogrammed version of the S2811 Signal Processing Peripheral. For further information on the internal operation of the S2811, please refer to the S2811 Advanced Product Description. It calculates FFTs and IFFTs using a decimation in frequency (DIF) technique for minimum distortion. The S2814A calculates a 32 complex point FFT using internally generated coefficients in a single pass. A coefficient rotation algorithm allows larger FFTs to be implemented (in blocks of 32 points). This implementation may be carried out by successive passes of the data through the two main routines in the S2814A, allowing larger transforms to be carried out with a single S2814A. Alternatively, an array of S2814As may be used to increase the transformation speed by parallel processing.

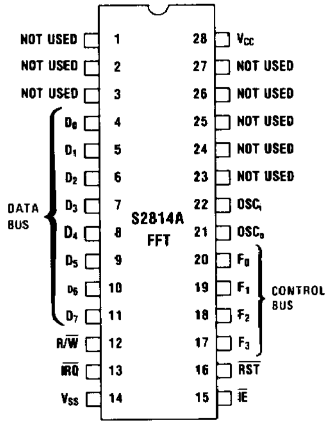
The word length used in the S2814A gives the transformed data a resolution of up to 57dB, but the total dynamic range can be increased up to 70dB by using the Conditional Array Scaling (CAS) routine incorporated.

The S2814A is intended to be used in a microprocessor system (see Block Diagram), using an 8 or 16 bit microprocessor, ROM, RAM and an optional DMA Controller or Address Generator. The S2814A is used as a

Block Diagram: Minimum System Configuration



Pin Configuration



memory mapped peripheral, and should be assigned a block of 16 addresses. It is used as a "hardware subroutine" function. The microprocessor controls the flow of data, including I/O, and calls the routines in the S2814A to cause the FFT to be executed. The S2814A responds to the microprocessor with the $\overline{IRQ}$ line when the processing of each routine is completed. In the case of a 32 point transform this signifies the completion of the transform, and in larger transforms it signifies that the microprocessor should unload the output data, load the next input data and call the next routine to be executed. The data is stored externally in RAM. Input data to be transformed is loaded into displacements 0 and 1 of the S2814A data memory. At the end of the FFT routine output data overwrites the input data. If power spectrum flag (PSF) is set, the S2814A computes the sum of the squares of the real and imaginary components of the output data and places the result in displacement 3 of the data memory. Both complex FFT data and power spectrum data are thus available. Windowing weights may be loaded into the S2814A prior to processing if the windowing routine is to be used. A 6800 compatible source listing of a suitable control program is available to the S2814A user at no charge. This control program will also be made available as a mask programmed ROM.

Absolute Maximum Ratings

Supply Voltage	7.0VDC
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	−55°C to +125°C
Voltage at any Pin	V_{SS} −0.3 to V_{CC} +0.3V
Lead Temperature (soldering, 10sec.)	200°C

Electrical Specifications (V_{CC}=5.0V±5%; V_{SS}=0V, T_A=0°C to 70°C unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
V_{IH}	Input High Logic "1" Voltage	2.0		V_{CC}+0.3	V	V_{CC}=5.0V
V_{IL}	Input LOW Logic "0" Voltage	−0.3		0.8	V	V_{CC}=5.0V
I_{IN}	Input Logic Leakage Current		1.0	2.5	μA	V_{IN}=0V to 5.25V
C_I	Input Capacitance			7.5	pF	
V_{OH}	Output HIGH Voltage	2.4			V	I_{LOAD}=−100μA, V_{CC}=min, C_L=30pF
V_{OL}	Output LOW Voltage			0.4	V	I_{LOAD}=1.6mA, V_{CC}=min, C_L=30pF
f_{CLK} (max)	Maximum Clock Frequency				MHz	V_{CC}=5.0V
	S2814A-10	10				
	S2814A-12	12				
	S2814A-15	15				
	S2814A	20				
P_D	Power Dissipation		1.2		W	V_{CC}=5.0V

S2814A Pin Functions/Descriptions

Pin	Number	Function
D_0-D_7	4-11	(Input/Output) Bi-directional 8-bit data bus. Data is Two's Complement coded.
F_0-F_3	20-17	(Input) Control Function bus. Four Microprocessor address lines (typically A_0-A_3) are used to control the S2814A.
$\overline{IE}$	15	(Input) Interface Enable. A low level on this line enables data transfer on the data bus and control functions on the F-bus. Usually generated by microprocessor address decode logic.
$R/\overline{W}$	12	(Input) Read/write select. When HIGH, output data from the S2814A may be read, and when LOW data may be written into the S2814.
$\overline{IRQ}$	13	(Output) Interrupt Request. This open drain output goes low when the S2814A has completed the execution of a routine and output data is available.
$\overline{RST}$	16	(Input) When LOW all registers and counters will be cleared, including the program counter, and all control functions cleared.
OSC_i, OSC_0	22,21	Oscillator input and output. For normal operation a crystal is connected between these pins to generate the internal clock signals. Alternatively, an external square wave signal may be connected to OSC_0 pin with OSC_i pin left open. All timings shown in this Product Description assume a 20MHz clock frequency.
V_{CC}	28	Positive power supply connection.
V_{SS}	14	Negative power supply connection. Normally connected to ground.

In addition to the above, pins 23-27 and 1 are connected internally. They should all be tied to V_{SS} during normal operation. Do not make connections to pins 2 and 3.

Functional Description

The S2814A is a pre-programmed version of AMI's S2811 Signal Processing Peripheral. This is a high speed microcomputer organized for efficient signal processing and contains a data memory, instruction memory, an arithmetic unit incorporating a 12-bit parallel multiplier, as well as control registers and counters. For more detailed information about the chip, please refer to the S2811 Advanced Product Description.

The S2814A Instruction ROM contains the various routines which make up the FFT package. The routines together with their starting addresses in the Instruction ROM, are shown in Table 1A.

The Data ROM contains the coefficients required to execute the functions. 128 words of Data RAM are provided to hold the 32 point complex signal data during processing as well as the power spectrum of the output and various other parameters, including the total number of points in the desired transform. The memory is organized as a 32x4 matrix, with the data arranged in columns, as shown in Table 1B.

Table 1: Software Model of S2814A

A. Routine Locations in Instruction Memory

LOC (HEX)	FUNCTION
00	IDLE STATE
01	ENTRY PT. "INIT" ROUTINE
04	ENTRY PT. "FFT32" ROUTINE
D3	ENTRY PT. "COMPAS" ROUTINE
EA	ENTRY PT. "SCALE" ROUTINE
DC	ENTRY PT. "WINDOW" ROUTINE
E4	ENTRY PT. "CONJUG" ROUTINE

B. Data Memory Map

(Note: Address [Base AB, Displacement C] is written as AB·C)

DISPLACEMENT BASE	0	1	2	3	4 5 6 7
00	REAL DATA (32 POINTS)	IMAGINARY DATA (32 POINTS)	ΔWORD	WINDOW FUNCTION (I/P) POWER SPECTRUM (O/P) (32 POINTS)	COEFFICIENT ROM
01			ΔSTEP		
02			NT		
03			SCIN		
04			CASEN		
05			PSF		
06			SCOUT		
•					
•					
•					
•					
1F					

C. Control Functions

F-BUS (HEX)	MNEMONIC	FUNCTION
1	RST	RESETS CHIP
2	DUH	SELECTS MSBYTE
3	DLH	SELECTS LSBYTE
4	XEQ	STARTS EXECUTION
9	BLK	SELECTS BLOCK MODE

D. Input and Output Registers

15 8	7 0	
DUH (MSBYTE)	DLH (LSBYTE)	INPUT REGISTER

15 8	7 0	
DUH (MSBYTE)	DLH (LSBYTE)	OUTPUT REGISTER

CODE IS TWO'S COMPLEMENT.

Initial Set-Up Procedure

After power up, the $\overline{RST}$ line should be held low for a minimum of 1 instruction cycle. If this line is connected to the reset line of the microprocessor this condition will be met easily. This will clear the Base and Index Registers, which are used for memory addressing, the Loop Counter and the Program Counter. Address zero in the Instruction ROM contains a Jump to Zero instruction, and thus the S2814A will remain in an idle state after being reset. Every routine in the memory is also terminated with a Jump to Zero instruction, and thus the S2814A will also remain in this same idle state after the the execution of each routine. The $\overline{IRQ}$ line will signal this condition each time, except after the initial reset and after execution of the INIT routine.

The Control Functions

The S2814A is controlled by the host microprocessor by means of the F-bus, Interface Enable ($\overline{IE}$) and the Read-Write (R/$\overline{W}$) lines. It should be connected to the host processor as a memory mapped peripheral as shown in Figure 1.

The 12 most significant address lines decode a group of 16 addresses to activate the $\overline{IE}$ line each time an address in the group is called, and the S2814A is controlled by reading to or writing from those addresses. Only 5 of these addresses are used as described in Table 2. Throughout this Product Description these addresses will be referred to as HHHX (X=0-F).

Figure 1. Connection of S2814A as a Memory Mapped Peripheral

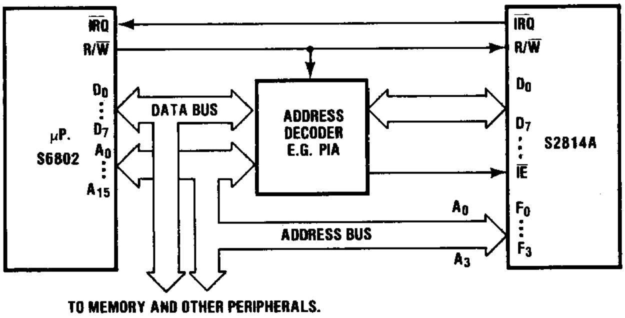


Table 2: S2814A Control Functions

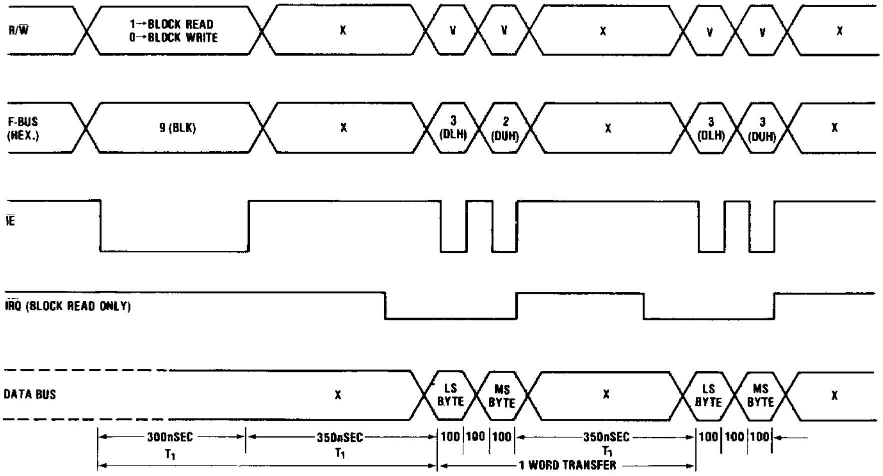
MNEMONIC	F-BUS HEX	DATA	TYPE OF OPERATION	FUNCTION
RST	1	XX	READ/ WRITE	CLEARS ALL REGISTERS. STARTS PROGRAM EXECUTION AT LOCATION 00. THIS IS THE IDLE STATE. THIS INSTRUCTION SHOULD PRECEDE BLOCK READ, BLOCK WRITE AND EXECUTE COMMANDS.
DUH	2	HH	READ/ WRITE	READS FROM OR WRITES INTO S2814A THE UPPER HALF OF THE DATA WORD. (SEE TABLE 1.D.)
DLH	3	HH	READ/ WRITE	READS FROM OR WRITES INTO S2814A THE LOWER HALF OF THE DATA WORD. (SEE TABLE 1.D)
XEQ	4	HH	WRITE	STARTS EXECUTION AT LOCATION HH
BLK	9	XX	READ/ WRITE	INITIATES A BLOCK READ OR BLOCK WRITE OPERATION. THE ENTIRE DATA RAM CAN BE ACCESSED SEQUENTIALLY BEGINNING WITH VALUES OF BASE AND DISPLACEMENT INITIALIZED USING "BLOCK TRANSFER SET UP" ROUTINE. IF A RESET OPERATION IS PERFORMED PRIOR TO BLOCK COMMAND THE DATA MEMORY ADDRESS IS INITIALIZED TO BASE 0, DISPLACEMENT 0. BLOCK READ OR WRITE OPERATION CAN BE TERMINATED ANY TIME BY PERFORMING A RESET OPERATION. THE INDEX REGISTER IS USED TO ADDRESS THE MEMORY DURING BLOCK TRANSFER AND INTERNAL ADDRESSING IS SEQUENCED AUTOMATICALLY.

NOTE: XX = Don't care
HH = 2 Hex characters (8-bit data)

The Block Transfer Operation

Block transfer is the mode used to load and unload the main data blocks into the S2814A at up to 4Mbytes/sec. In this mode the data memory is addressed by the Index Register, and after initialization the internal addressing is sequential and automatic. The sequence generated is that after each word transfer (16-bit words as 2 bytes, or 8-bit words as MSbyte (DUH) only) the base is incremented. After base 1F (31) has been reached, the base resets to 00 and the displacement increments. After base 1F displacement 3 has been reached (i.e., the highest address in the RAM, 1F.3), both base and displacement reset to zero. Note that when the BLK command is given the Read/Write line is latched internally, and remains latched until the RST command is given. The block transfer sequence and timing are shown in Figure 2. When using a microprocessor to execute the block read it will normally be advantageous to set the interrupt mask.

Figure 2. Block Transfer Sequence and Timing



NOTE 1: X = DON'T CARE, OR NOT VALID V = VALID

NOTE 2: ALL TIMES SHOWN ARE MINIMUM AND MUST BE INCREASED PROPORTIONALLY WHEN USING REDUCED SPECIFICATION PARTS AS FOLLOWS:

$T_{1\,MIN} = \left[\frac{6000}{f_{CLK}} + 50\right]$ nsec

In 6800 Assembly Language a Block Write would be executed with the following code:

```
LDX        OFFST   ;LOAD MEMORY START AD-
                    DRESS INTO INDEX REG.
STA    A   BLK     ;WRITE DUMMY DATA TO AD-
                    DRESS $HHH9,BLOCK MODE.
LDA    A   0,X     ;READ FIRST BYTE FROM
                    MEMORY.
STA    A   DLH     ;WRITE INTO S2814A AS LSBYTE.
                    ADDRESS $HHH3
LDA    A   1,X     ;READ SECOND BYTE FROM
                    MEMORY.
STA    A   DUH     ;WRITE INTO S2814A AS
                    MSBYTE.ADDRESS $HHH2
LDA    A   2,X     ;SECOND WORD.
 :     :            :
 :     :            :
LDA    A   62,X    ;32ND. WORD.LSBYTE.
STA    A   DLH     ;
LDA    A   63,X    ;32ND. WORD,MSBYTE.
STA    A   DUH     ;END OF TRANSFER.
STA    A   RST     ;WRITE DUMMY DATA TO AD-
                    DRESS $HHH1.RESET.
```

Block Read would be executed by substituting LDA A for STA A, and vice versa.

where:

```
RST   EQU  $HHH1
DLH   EQU  $HHH3
DUH   EQU  $HHH2
BLK   EQU  $HHH9
```

The above code assumes that the block transfer is controlled by the host processor, not using DMA. Note that DLH must always precede DUH. 8-bit data may be transferred using DUH only, assuming that the significance of the data is correct.

The FFT Routines

Six individual routines are stored in the S2814A Instruction memory. Two or more of these are used in the computation of an FFT, depending on the transform size and the options selected. The starting addresses of the routines are shown in Table 3. All execution times quoted assume a 20MHz clock frequency.

Table 3. FFT Routines and Their Starting Addresses

LOCATION (HEX)	FUNCTION
00	IDLE STATE
01	ENTRY POINT FOR "INIT" ROUTINE
	(IR) = BASE, DISPLACEMENT
	$(BASE)_{4\text{-}0} \leftarrow (IR)_{15\text{-}11}, (DISP)_{1,0} \leftarrow (IR)_{9,8}$
	Returns to Idle state Exec. Time = 0.9μs
04	ENTRY POINT FOR "FFT32" ROUTINE
	(DISP0) = Input Data (Real), (DISP1) = Input Data (Imag.) (DISP2) = SCIN, CASEN, PSF
	Perform 32 point FFT. Sets IRQ, Returns to Idle state. Exec. Time = 1.2 ms to 1.8ms.
	(OR) = SCOUT (DISP0) = Transformed Data (Real), (DISP1) = Transformed Data (Imag.) (DISP2) = SCOUT, (DISP3) = Power Spectrum Data if PSF = 1
D3	ENTRY POINT FOR "COMPAS" ROUTINE
	(DISP0) = Input Data (Real), (DISP1) = Input Data (Imag.) (DISP2) = WORD, STEP, NT, SCIN, CASEN
	Perform COMPAS, Sets IRQ, Returns to Idle State Exec. Time = 233 to 374μsec.
	(DISP0) = Output Data (Real), (DISP1) = Output Data (Imag.) (DISP2) = SCOUT, (OR) = SCOUT
EA	ENTRY POINT FOR "SCALE" ROUTINE
	(IR) = SCLP, (DISP0) = Data (Real), (DISP1) = Data(Imag.)
	Performs scaling, Sets IRQ. Returns to Idle State Exec. Time = 51 to 250μsec.
	(DISP0) = Scaled Data (Real), (DISP1) = Scaled Data (Imag.)
DC	ENTRY POINT FOR "WINDOW" ROUTINE
	(DISP0) = Input Data (Real), (DISP1) = Input Data (Imag.) (DISP3) = Multiplying factors
	Performs multiplication, Sets IRQ, Returns to Idle State Exec. Time = 49μsec.
	(DISP0) = Output Data (Real), (DISP1) = Output Data (Imag.)
E4	ENTRY POINT FOR "CONJUG" ROUTINE
	No set-up required. Conjugates input data (negates imaginary components). Sets IRQ. Returns to Idle State. Exec. time = 30μsec.

1. Block Transfer Set-Up (INIT). Entry Address 01.

This routine presets the Index Register to allow block transfer to commence at any location other than 00.0 in the S2814A data RAM. An eight-bit word is loaded into the upper half of the input register and the routine executed as shown:

Figure 3A. Flowchart for Subroutine FT32IN

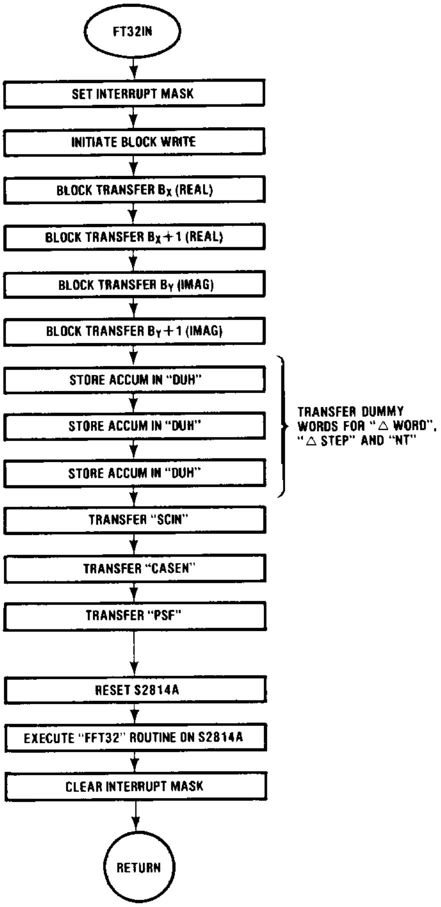
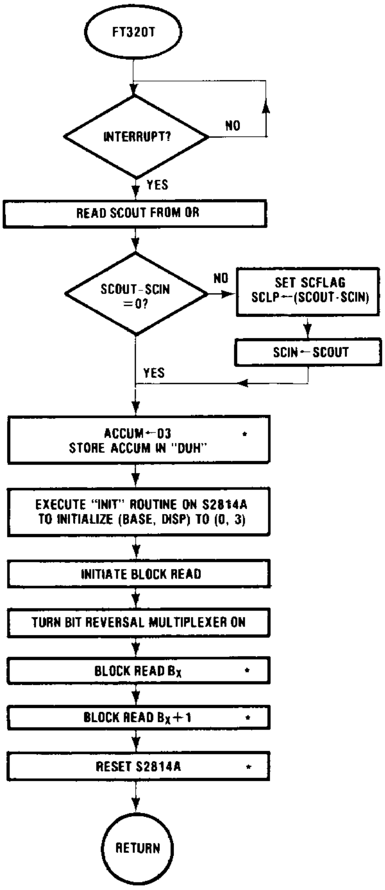

FT32IN
SET INTERRUPT MASK
INITIATE BLOCK WRITE
BLOCK TRANSFER B_X (REAL)
BLOCK TRANSFER B_X+1 (REAL)
BLOCK TRANSFER B_Y (IMAG)
BLOCK TRANSFER B_Y+1 (IMAG)
STORE ACCUM IN "DUH"
STORE ACCUM IN "DUH"
STORE ACCUM IN "DUH"
TRANSFER DUMMY WORDS FOR "△ WORD", "△ STEP" AND "NT"
TRANSFER "SCIN"
TRANSFER "CASEN"
TRANSFER "PSF"
RESET S2814A
EXECUTE "FFT32" ROUTINE ON S2814A
CLEAR INTERRUPT MASK
RETURN

Figure 3B. Flowchart for Subroutine FT320T



*ASSUMES POWER SPECTRUM ONLY IS READ OUT.

```
DUH   EQU  $HHH2
XEQ   EQU  $HHH4


LDA   A  #$XX
STA   A  DUH
LDA   A  #1
STA   A  XEQ
```

where XX represents the start address for block transfer. The routine will be executed in 3 instruction cycles (0.9μsec.) and the S2814A will return to the idle state. Block transfer may then commence immediately.

2. FFT32. Entry Address = 04.

This is the basic 32 complex point FFT routine. For a 32 point FFT this routine is called once only and the output of the routine is the FFT. Larger FFTs are computed by decimating them into 32 point arrays before final processing of these arrays using FFT32 to obtain the final outputs. The following data is loaded into the S2814A, using block write starting at address 00.0, i.e., INIT is not used.

32 words of real input data (addresses 00.0 - 1F.0)

32 words of imaginary input data (addresses 00.1 -1F.1)

3 dummy words (to skip addresses) (addresses 00.2 -02.2)

SCIN (input scaling parameter) (address 03.2)

CASEN (CAS Enable) (address 04.2)

PSF (Power spectrum flag) (address 05.2)

Note that CASEN (Conditional array scaling enable) and PSF are not modified during processing, and need only be loaded once. CASEN should be positive to inhibit CAS (e.g. 0000) and negative to enable CAS (e.g. 8000). Note that SCIN is not needed if CAS is not enabled. PSF should be zero if the power spectrum output is not needed, any non-zero value (e.g. 0100) will cause the power spectrum to be computed. The block transfer should be terminated with the RST command, and the FFT32 routine called. Flow charts for loading and dumping the data are shown in Figure 3. The following sequence will cause the execution of the entire function:

```
CLR  B          ;CLEAR B ACC.
STA  A RST      ;RESET S2814A REGISTERS.
SEI             ;SET INT. MASK.
STA  A BLK      ;SET UP BLOCK WRITE.
JSR    BLKWT    ;WRITE 64 WORDS OF DATA.
STA  A DUH      ;WRITE DUMMY DATA TO 00.0
STA  A DUH      ;.....................TO 00.1
STA  A DUH      ;.....................TO 00.2
LDA  A SCIN     ;FETCH SCIN.
STA  A DLH      ;WRITE TO ADDRESS 00.3
STA  B DUH      ;COMPLETE WORD XFER.
LDA  A CASEN    ;FETCH CAS ENABLE.
STA  A DUH      ;WRITE TO ADDRESS 00.4
LDA  A PSF      ;FETCH PS FLAG.
STA  A DUH      ;WRITE TO ADDRESS 00.5
STA  A RST      ;RESET S2814A.
LDA  A #4       ;FFT32 START ADDRESS.
STA  A XEQ      ;START EXECUTING.
CLI             ;CLEAR INT. MASK.
WAI             ;WAIT FOR ROUTINE END.
LDA  A DLH      ;START OF INT. ROUTINE.
LDA  B DUH      ;(DUMMY).READ SCOUT.
LDA  B SCIN     ;FETCH SCIN.
STA  A SCIN     ;SCOUT→SCIN
SBA             ;COMP.SCOUT WITH SCIN.
BEQ    READ     ;JUMP IF NO CHANGE.
STA  A SCLP     ;(SCOUT-SCIN) → SCLP
LDA  A PASSN    ;FETCH PASS #
CMP  A #1       ;IS THIS 1ST.PASS?
BEQ    READ     ;IF SO, JUMP
JSR    SKOUT    ;SCALE PREVIOUS ARRAYS
LDA  A #3       ;(ASSUME PSF SET
STA  A DUH      ;PRESET TO ADDRESS 00.3
LDA  A #1       ;
STA  A XEQ      ;EXECUTE INIT.
STA  A BRV      ;TURN ON BIT REV.MUX.
LDA  A BLK      ;SET UP BLOCK READ.
JSR    BLKRD    ;READ DATA.
STA  A RST      ;END
```

The routine execution time is variable, depending on whether CASEN and PSF are set. The times are:

1. CAS - OFF. PSF - OFF 3730 instruction cycles (1.119msec.)
2. CAS - OFF. PSF - ON 3862 instruction cycles (1.159msec.)
3. CAS - ON . PSF - OFF 5867max. instruction cycles (1.760msec.)
4. CAS - ON . PSF - ON 5999max. instruction cycles (1.800msec.)

When CAS is enabled, the time depends on the number of times overflow is corrected. At the end of the routine the complex output data will have overwritten the input data in the memory (addresses 00.0 to 1F.1) and the power spectrum data will be in displacement 3 (addresses 00.3 -1F.3). The output scaling factor (SCOUT) will be loaded in the output register, generating the $\overline{\text{IRQ}}$ to signify to the host processor that the routine has completed processing.

3. Combination Pass Routine, COMPAS. Entry Address = D3.

This is the decimation routine that breaks up larger transforms into a number of 32 point transforms to be executed by FFT32. The N data points are split into N/16 blocks of 16 points, and pairs of blocks are passed through COMPAS. The procedure is repeated one or more times if N is greater than 64, but for a 64 point FFT the resulting data is ready for processing using FFT32. The procedure is explained in greater detail in the section "Executing Larger Transforms". The following data is loaded into the S2814A before execution:

Figure 4A. Flowchart for Subroutine CSIN

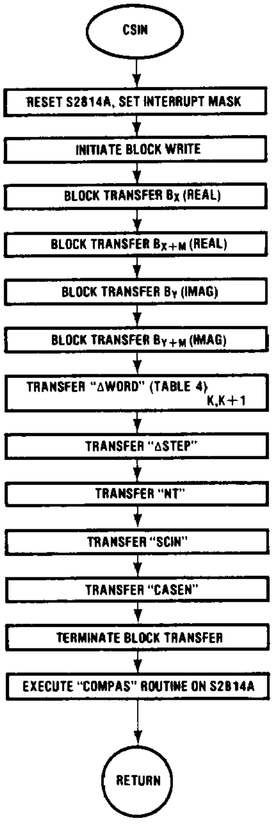
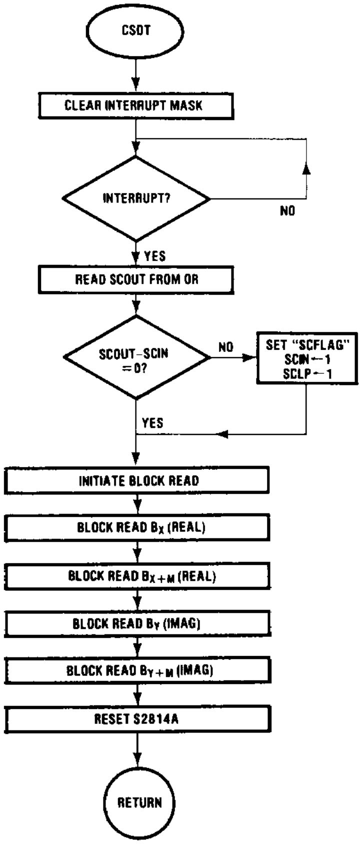

CSIN
RESET S2814A, SET INTERRUPT MASK
INITIATE BLOCK WRITE
BLOCK TRANSFER B_X (REAL)
BLOCK TRANSFER B_{X+M} (REAL)
BLOCK TRANSFER B_Y (IMAG)
BLOCK TRANSFER B_{Y+M} (IMAG)
TRANSFER "ΔWORD" (TABLE 4) K,K+1
TRANSFER "ΔSTEP"
TRANSFER "NT"
TRANSFER "SCIN"
TRANSFER "CASEN"
TERMINATE BLOCK TRANSFER
EXECUTE "COMPAS" ROUTINE ON S2814A
RETURN

Figure 4B. Flowchart for Subroutine CSOT



32 words of real input data (addresses 00.0 - 1F.0)

32 words of imaginary input data (addresses 00.1 -1F.1)

Δ WORD (address 00.2)

Δ STEP Set up parameters (address 01.2)

NT (address 02.2)

SCIN (address 03.2)

CASEN (address 04.2)

PSF (address 05.2)

The new parameters required, Δ WORD, Δ STEP and NT are dependent on the size of the transform and Δ WORD changes with each pass through the COMPAS routine. The values required are shown in the tables in sections "Executing 64 Point Transforms" and "Executing Larger Transforms". Flow charts for loading and dumping the data are shown in Figure 4. The routine execution time varies with transform size and depends on whether CAS is enabled or not, as shown:

TRANSFORM SIZE	64 POINT	128 POINT	256 POINT	512 POINT
Without CAS, Inst. cycles, (μsec.)	776 (233)	828 (248)	842 (253)	949 (255)
With CAS. (Max.) Inst. cycles (μsec.)	1172(352)	1224(367)	1238(371)	1245(374)

4. Data Point Scaling Routine, SCALE. Entry location = EA.

If CAS is enabled, then routines COMPAS, and FFT32 will scale all 32 data points being processed if an overflow occurs during that pass. The value of SCOUT allows the data during subsequent passes to be scaled automatically during the pass. However, data points which have already been processed must also be scaled, so that all the data is scaled by the same factor during each processing step. SCALE is a routine that allows this to be done at high speed. Each block to be scaled is block loaded into the S2814A, the routine SCALE executed, and the block dumped back into the original locations in memory.

Care must be taken to keep track of which blocks have already been processed during each step, so that blocks do not get missed or scaled twice. The execution time depends on the scaling factor (SCOUT), as shown below:

Scaling Factor (SCOUT)	1	2	3	4	5
Execution time. Inst. Cycles, (μsec.)	170(51)	336(101)	502(151)	668(200)	834(250)

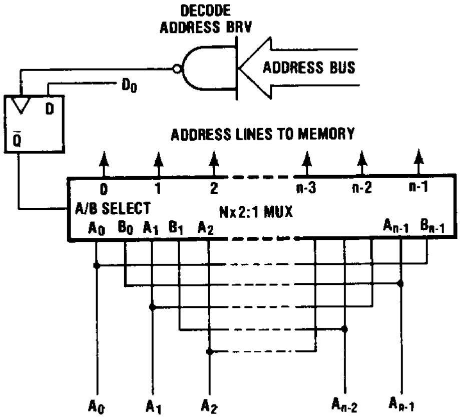
Windowing Routine, WINDOW. Entry Address = DC.

In order to allow the input data points to be windowed, a routine is provided to multiply the 32 real or complex points loaded in the S2814A by 32 window points. This is done on each block of 32 points prior to commencing the actual FFT processing. The input data required, in addition to the normal input data, are the 32 points of the window. They should be loaded into displacement 3 of the S2814A RAM and the routine WINDOW executed. The windowed data points will be returned to their original positions in the memory, so that COMPAS or FFT32 may then be executed immediately without further processing. The entire data can be loaded in a single block transfer operation by using INIT to preset the start address to 00.3. The 32 point window data is then loaded, followed by the signal data. This is possible because after loading the window the memory address will automatically reset to 00.0, the start address for the real data. The parameters are then loaded into displacement 2 addresses in the usual way. They will not be affected by the windowing operation. The total execution time is 163 instruction cycles, 49μsec.

Executing FFTs

Executing the FFTs consists of loading data blocks, executing routines in the S2814A and dumping the data. However, the sequence of the FFT output data is scrambled, and in order to use the results meaningfully, it must be unscrambled. This is done by reversing the order of the bits of the address lines for the final output data. Thus, for a 2^N point FFT the N address lines A_0, A_1, A_2 ..... A_{N-1} must be reversed to the sequence A_{N-1}, A_{N-2} ..... A_1, A_0 to address the output buffer memory. This is most conveniently done as the data points are being dumped out of the S2814A after the processing of the FFT32 routine(s). The bit reversal can be done either by software or hardware. The hardware realization is shown in Figure 5, and an example of software bit reversal is given in the section "Executing 32 Point FFTs."

Figure 5. Bit Reversal Hardware



WRITING A 1 TO ADDRESS BRV TURNS ON BIT-REVERSAL
WRITING A 0 CLEARS BIT REVERSAL

Executing 32 Point Transforms

The basic 32 point transform is easily implemented with the S2814A since it simply requires the loading of the 32 real or complex data points and the 3 parameters SCIN, CASEN and PSF, executing FFT32 once only and dumping the data using bit reversal. The flowchart for this sequence is shown in Figure 6. It is assumed that the loading of data from the source into the input buffer and dumping of data from the output buffer to destination is carried out by the NMI (non-maskable interrupt) routine. The parameter SCIN should be set to zero, and the output data should be scaled (multiplied) by $2^{(SCOUT)}$ if absolute levels are wanted.

Figure 6. Flowchart for 32 Point FFT

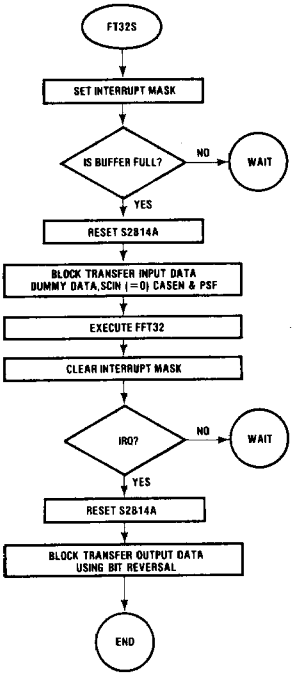
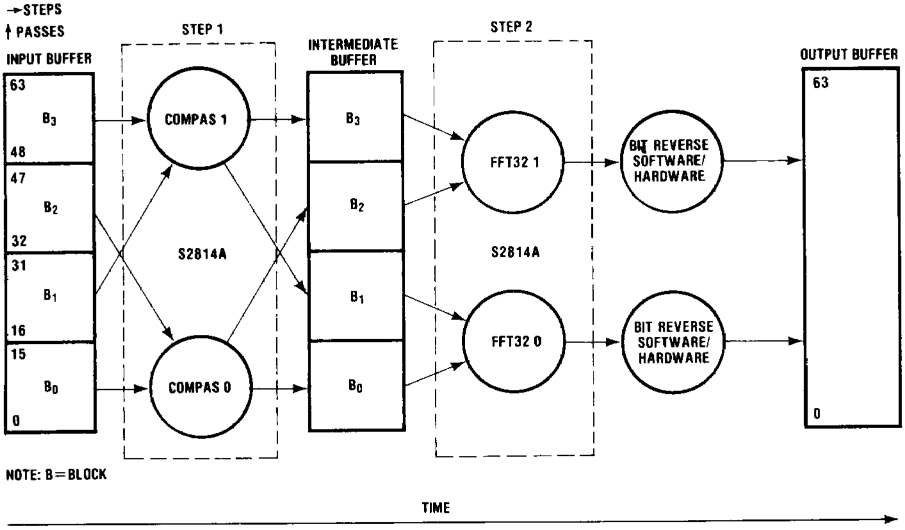


Figure 7. 64 Point FFT Flowgraph



Executing 64 Point Transforms

This is the simplest expansion of the FFT. The first step is to use COMPAS (twice) to decimate the data into two 32 point transforms, and then use FFT32 (twice) to produce the transforms. This is shown in the signal flow graph in Figure 7. The flow graph is independent of whether one or two S2814As are used, since the two passes through each of the 2 routines (COMPAS and FFT32) can be carried out sequentially or in parallel. The set up parameters for the 64 point FFT are:

$$\left.\begin{array}{l}\text{For COMPAS 0: }\Delta\text{WORD}=8070\\ \text{For COMPAS 1: }\Delta\text{WORD}=\text{C070}\end{array}\right\}\Delta\text{STEP}=4000\ \text{NT}=1$$

The treatment of SCIN and SCOUT is dealt with in the next section.

Executing Larger Transforms

The execution of larger transforms follows the same sequence as the 64 point transforms; namely the decimation of the data into a series of 32 point blocks that can be processed using FFT32. For a 2^N point FFT this involves N-5 steps of processing using COMPAS, and each step requires $2^{(N-5)}$ passes through the COMPAS routine. This is followed by $2^{(N-5)}$ passes through the FFT32 routine. Within each step, each pass may be carried out sequentially using a single S2814A, or in parallel using $2^{(N-5)}$ chips. There are also intermediate sequential + parallel combinations possible, of course, using fewer chips. A signal flow graph for 1 step is shown in Figure 8.

At the start of each step, SCIN should be set to zero. For the remaining passes in that step the value of SCOUT for the current pass should be used for SCIN for the next pass. The outputs of previously computed passes must be scaled using routine SCALE each time SCOUT increases during a pass. The maximum value of SCOUT after executing COMPAS is 1, and after executing FFT32 it is 5.

A flow chart for an N point transform control program is shown in Figure 9. The routine is called NFFT and uses the following subroutines.:

CSIN — procedure for loading S2814A with COMPAS input data (Figure 4A)

CSOT — procedure for dumping COMPAS output data (Figure 4B)

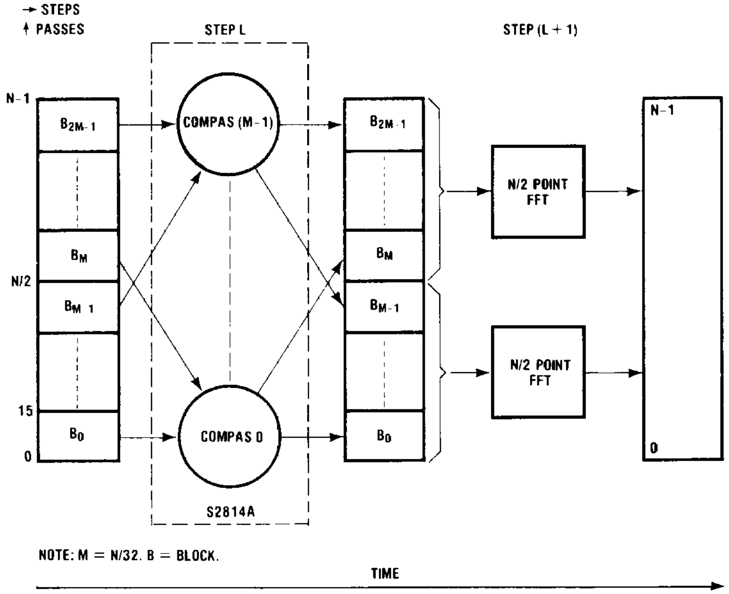
SCLPRV — procedure for scaling previously computed blocks of data in each step. See Figure 10.

FT32IN — procedure for loading S2814A with FFT32 input data (Figure 3a)

FT32OT — procedure for dumping FFT32 output data. (Figure 3b)

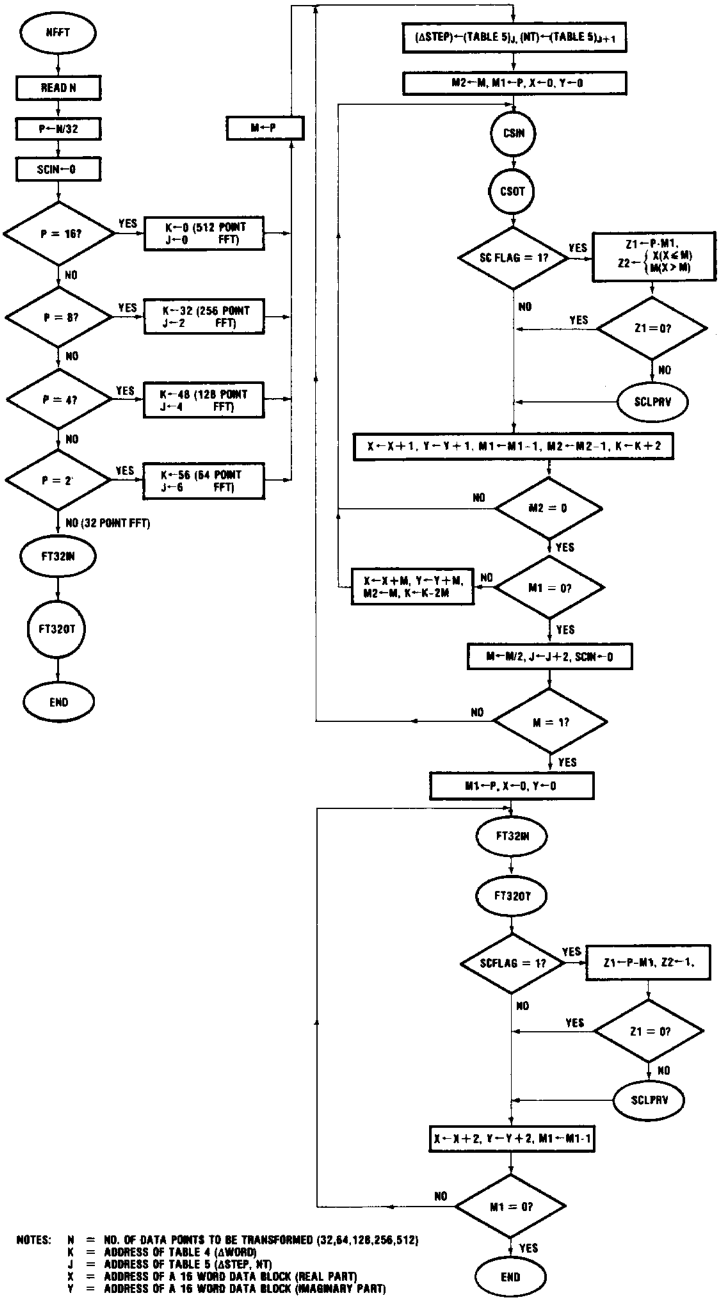
The values of ΔWORD, ΔSTEP and NT are shown in Tables 4 and 5.

Figure 8. N Point FFT Flowgraph



NOTE: M = N/32. B = BLOCK.

Figure 9. Flow Chart for N Point FFT, Routine "NFFT"



NOTES:
N = NO. OF DATA POINTS TO BE TRANSFORMED (32,64,128,256,512)
K = ADDRESS OF TABLE 4 (ΔWORD)
J = ADDRESS OF TABLE 5 (ΔSTEP, NT)
X = ADDRESS OF A 16 WORD DATA BLOCK (REAL PART)
Y = ADDRESS OF A 16 WORD DATA BLOCK (IMAGINARY PART)

Figure 10. Flowchart for Subroutine "SCLPRV"

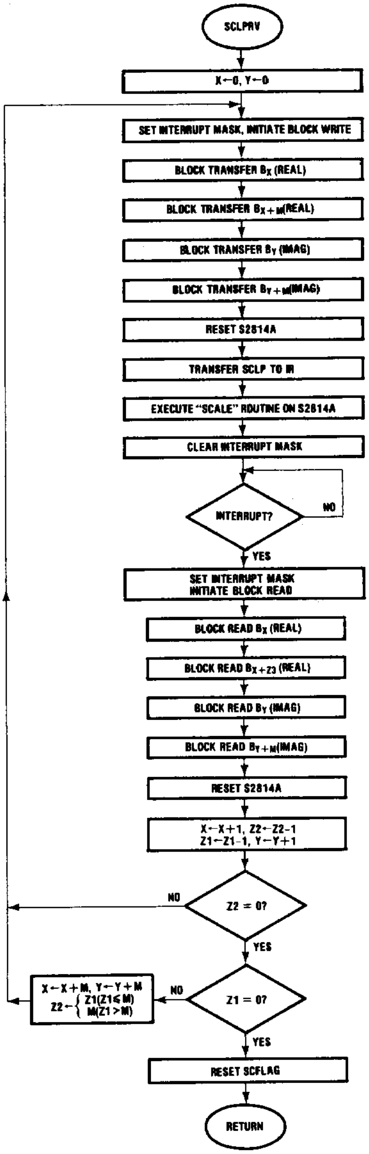

SCLPRV
X←0, Y←0
SET INTERRUPT MASK, INITIATE BLOCK WRITE
BLOCK TRANSFER B_X (REAL)
BLOCK TRANSFER B_{X+M}(REAL)
BLOCK TRANSFER B_Y (IMAG)
BLOCK TRANSFER B_{Y+M}(IMAG)
RESET S2814A
TRANSFER SCLP TO IR
EXECUTE "SCALE" ROUTINE ON S2814A
CLEAR INTERRUPT MASK
INTERRUPT?
NO
YES
SET INTERRUPT MASK
INITIATE BLOCK READ
BLOCK READ B_X (REAL)
BLOCK READ B_{X+Z3} (REAL)
BLOCK READ B_Y (IMAG)
BLOCK READ B_{Y+M}(IMAG)
RESET S2814A
X←X+1, Z2←Z2−1
Z1←Z1−1, Y←Y+1
NO
Z2 = 0?
YES
X←X+M, Y←Y+M
$Z2 \leftarrow \begin{cases} Z1 (Z1 \leq M) \\ M (Z1 > M) \end{cases}$
NO
Z1 = 0?
YES
RESET SCFLAG
RETURN

Table 4. (ΔWORD)

ENTRY PT for	K	VALUE	COMMENTS
512→	0	00	(ΔWORD L)
point	1	80	(ΔWORD H)
x'form	2	00	
	3	88	
	4	00	
	5	90	
	6	00	
	7	98	
	8	00	
	9	A0	
	10	00	
	11	A8	
	12	00	
	13	B0	
	14	00	
	15	B8	
	16	00	
	17	C0	
	18	00	
	19	C8	
	20	00	
	21	D0	
	22	00	
	23	D8	
	24	00	
	25	E0	
	26	00	
	27	E8	
	28	00	
	29	F0	
	30	00	
	31	F8	
256→	32	10	
point	33	80	
x'form	34	10	
	35	90	
	36	10	

Table 4 (continued)

ENTRY PT for	K	VALUE
	37	A0
	38	10
	39	B0
	40	10
	41	C0
	42	10
	43	D0
	44	10
	45	E0
	46	10
	47	F0
128 → point x'form	48	30
	49	80
	50	30
	51	A0
	52	30
	53	C0
	54	30
	55	E0
64 → point x form	56	70
	57	80
	58	70
	59	C0

Table 5. (ΔSTEP, NT)

ENTRY PT for	J	VALUE	COMMENTS
512 point x'form	0	08	ΔSTEP(DUH)
	1	0F	NT(DLH)
256	2	10	''
	3	07	''
128	4	20	''
	5	03	''
64	6	40	''
	7	01	''

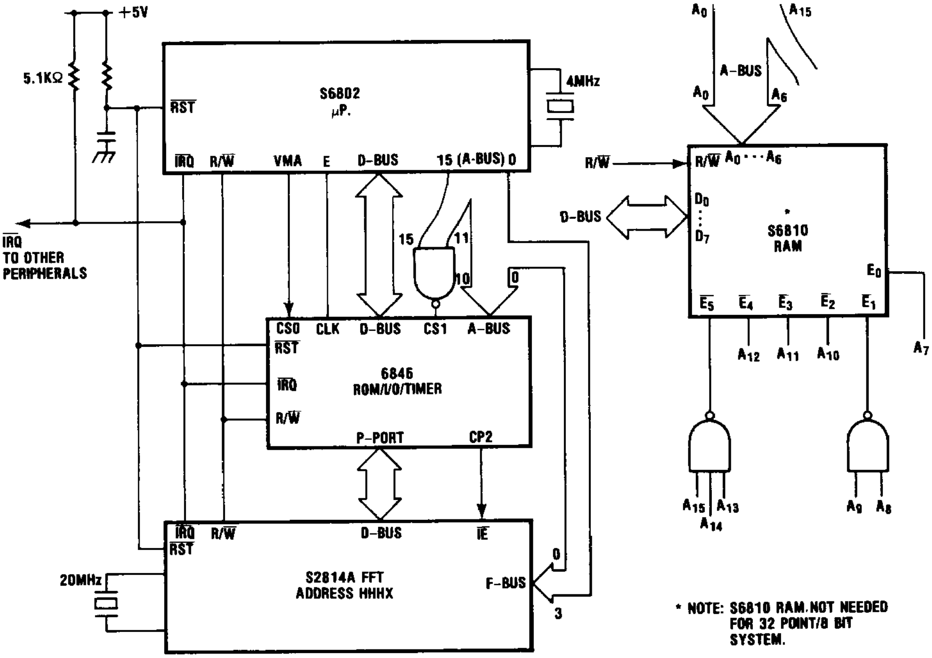
Hardware.

The minimum hardware for a 32 point FFT is shown in Figure 11. All data transfer and control is handled by the S6802. The availability of the next input sample is signalled with the NMI line. A suitable analog interface is shown in Figure 12. The sampling clock is derived from the microprocessor clock, and the NMI signal is generated by the EOC (end of conversion) output of the A/D converter. This system may be expanded simply by adding more memory. The memory requirements are shown in Table 6. A word may be up to 16 bits long. In order to speed up the complete procedure it is necessary to use DMA for block transfer of data. The S2814A will transfer data at up to 4Mbytes/sec. A suitable DMA Address Generator is the Advanced Micro Devices AM 2940, but a 68B44 will accomplish the function more conveniently at a slightly lower speed (1.5Mbyte/sec).

Data Bus Interface.

Figure 13 shows how to interface the S2814A with a typical 6800 family microprocessor data bus. Note that the S2814A data bus must be isolated from the microprocessor system data bus by use of a PIA as in Figure 11 or a 74LS245 or 74LS645 type data transceiver as shown in Figure 13, since the S2814A drive capability is only one TTL load. The bus isolation may be omitted in some small systems.

Figure 11. 32/64 Point FFT Hardware



* NOTE: S6810 RAM NOT NEEDED FOR 32 POINT/8 BIT SYSTEM.

Figure 12. Analog Interface

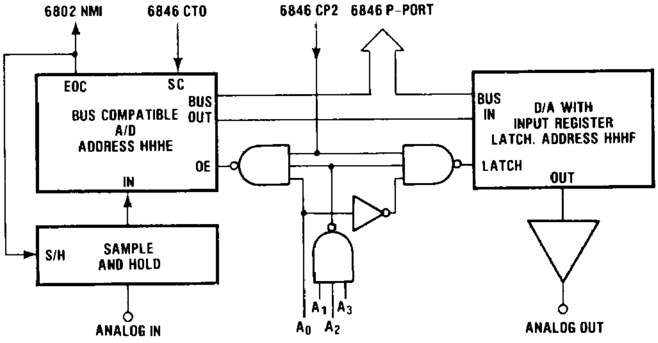


Figure 13. Interfacing the S2814A with a Microprocessor

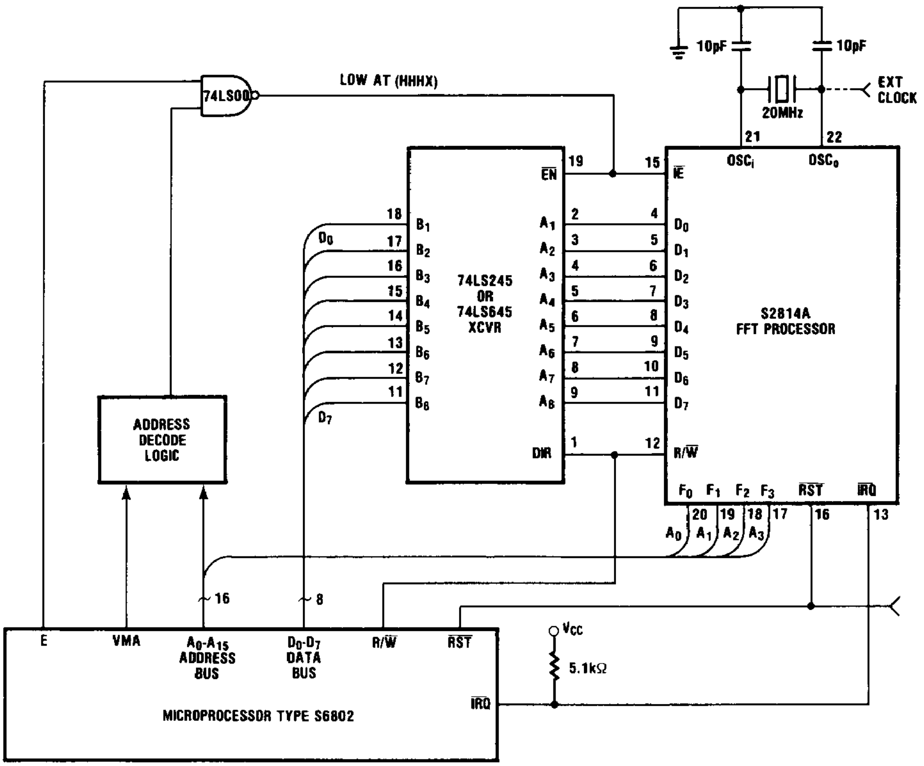


Table 6. Memory requirements for data point storage.

TRANSFORM SIZE (POINTS)	WORD LENGTH (BITS)	MEMORY REQUIREMENTS
32	8 10/12 16	64 bytes See Note 1 128 bytes
64	8 10/12 16	128 bytes See Note 1 256 bytes
128	8 10/12 16	256 bytes 768 nibbles 512 bytes
256	8 10/12 16	512 bytes 1536 nibbles 1024 bytes
512	8 10/12 16	1024 bytes 3072 nibbles 2048 bytes

Note 1: In practice the memory realization for these cases will be the same as for 16-bit systems.

Transform Execution Times.

The maximum execution times of transforms are shown in Table 7. The actual execution time when CAS is enabled will be between the times shown for CAS off and the maximum with CAS on. It will depend on the number of times that scaling has to be done.

Table 4. Total FFT execution times including block transfers. (msec.)

	USING SINGLE S2814A BLOCK TRANSFER USING:				USING MULTIPLE S2814A ARRAY		
TRANSFORM SIZE	A S6802 (22μsec/word)		B DMA 2MW/sec		# OF S2814As	C (USING DMA AT 2MW/sec)	
	MIN	MAX	MIN	MAX		MIN	MAX
32 pt.	4.0	4.6	1.3	1.9	1	1.3	1.9
64	14.2	15.7	3.2	4.6	2	1.6	2.3
128	40.7	44.0	7.6	11.0	4	1.9	2.8
256	106	114	17.8	25.4	8	2.3	3.2
512	262	280	40.7	57.9	16	2.6	3.7

Note: Minimum times assume that CAS and PSF are off. Maximum times assume that CAS and PSF are on, and that maximum overflow occurs during 1st pass. All times assume 20MHz clock frequency and must be increased proportionally for lower clock frequencies (except Column A).

FFT Resolution and Dynamic Range

The use of the Decimation in Frequency (DIF) algorithm in the S2814A ensures optimum signal to noise ratio, (SNR) for the architecture used. The use of the Conditional Array Scaling (CAS) gives a total dynamic range of approximately 70dB on all sizes of Transforms. The maximum resolution obtainable is approximately 57dB. CAS operates by detecting overflow in the butterfly computation routine. As soon as an overflow is detected the two points being combined in that butterfly are halved in magnitude (both the real and imaginary portions) and the butterfly recomputed. A flag is set, all previously computed butterfly outputs are then scaled, and all the inputs to subsequent butterflies are scaled before computation begins, so that at the end of the pass all points have been scaled equally. A scale factor is made available (SCOUT) so that the remaining data points in larger transforms, i.e., those other than the 32 in the S2814A when the overflow occurred, may also be scaled to keep them all in line. Thus, CAS operates as a discrete AGC, halving the signal levels each time an overflow is detected. By using SCOUT after executing the FFT the output may be expanded, so that the levels displayed in the spectrum will increase monotonically as the input increases.