

生體信號處理用 Bit-slice Signal Processor에 관한 研究

김영호* · 김동록* · 민병구*

=Abstract=

A Study on the Bit-slice Signal Processor for the Biological Signal Processing

Young-Ho Kim,* Dong-Lok Kim,* Byoung-Goo Min*

We have developed a microprogrammable signal processor for real-time ultrasonic signal processing. Processing speed was increased by the parallelism in horizontal microprogram using 104bits microcode and the pipelined architecture.

Control unit of the signal processor was designed by microprogrammed architecture and writable control store(WCS) which was interfaced with host computer, APPLE-II. This enables the processor to develop and simulate various digital signal processing algorithms. The performance of the processor was evaluated by the Fast Fourier Transform(FFT)program. The execution time to perform 16 bit 1024 points complex FFT, radix-2 DIT algorithm, was about 175 msec with 1MHz master clock.

We can use this processor to develop more efficient signal processing algorithms on the biological signal processing.

1. 서 론

1965년 Cooley-Tukey의 Fast Fourier Transform Algorithm이 발표된 이후¹⁾ 반도체기술의 발전으로 보다 빠르고 큰 용량의 각종 special purpose processor가 개발됨으로써, digital 신호처리 분야는 크게 발전하고 있다.

초기에는 FFT를 포함한 대부분의 algorithm이 범용 computer를 이용하여 software로 처리함으로써 속도가 느려서 실시간 처리가 불가능하며 비효율적이었다. 그 후 특수한 목적에 적합하도록 설계한 각종 processor들이 개발되어 digital 신호처리 algorithm들의

효율적인 수행을 위해 look-up table, pipelining, parallelism 등을 사용하여 계산속도를 증가시키고 있다.²⁾

본 논문에서는 최근 의용전자 부문에서 활발히 연구되고 있는 초음파 반사신호로부터 생체내 조직 특성변수를 추출하기 위하여 16bit signal processor를 개발하여 spectrum을 분석하고자 한다. 본 processor는 FFT를 위한 special purpose processor로 설계되었으나 writable control store (WCS)를 사용하여 microprogramming이 가능함으로써 다른 여러가지 digital 신호처리 algorithm들의 개발 및 수행이 가능하다.

2. 생체 신호처리 방법의 개요

(1) 생체내 조직 특성

생체내 조직 특성을 관찰하기 위하여 초음파 신호로부터 감쇄계수를 측정하여 분석하는 방법이 있다.³⁾ 초음파 투과파로부터 감쇄계수를 측정하는 경우 투과에

<1985. 10. 5 접수>

*서울대학교 의과대학 의공학과

*Department of Biomedical Engineering, College of Medicine, Seoul National University

너지는 크나 투과 불가능한 골격등의 영향으로 임상적 용은 부적당하다. 반사파에 의하여 측정하는 경우에 너지는 낮고 noise가 큰 단점이 있으나 여러가지 기법에 의하여 감쇄계수를 추정할 수 있다. 반사신호의 투과거리에 대한 중심주파수 편이값의 기울기에 의한 spectral shift 법, 임의의 매질에 대한 입사파와 반사파의 spectrum을 구하여 그 차분에 의하여 감쇄계수를 측정하는 spectral difference법등으로 signal의 주파수 특성을 살펴봄으로써 조직 특성변수를 구하는 방법이 널리 사용되어지고 있다.

(2) Spectrum 해석 방법의 개요

거의 모든 생체신호는 random time function으로서 이러한 신호의 spectral measurement는 여러가지 요소에 의하여 상당히 복잡하다. 그러므로 이러한 신호에 대한 유한한 갯수의 관측치로부터 그 신호의 spectrum을 추정하는 여러가지 algorithm이 있다⁴⁾.

이 spectrum estimation은 parametric method와 nonparametric method의 2가지로 크게 나눌 수 있으며, parametric method에는 autoregressive(AR) model, moving average(MA) model, autoregressive moving average(ARMA) model에 의한 방법 및 maximum entropy method(MEM) 등이 있다. Non-parametric method에는 periodogram, Blackman-Tukey 및 각종 FFT algorithm들이 있으며, 본 processor에서는 우선 여러가지 FFT algorithm들을 수행할 수 있도록 hardware를 구성한 다음 microprogram을 작성하여 spectrum을 효율적으로 관찰할 수

있도록 한다.

3. Signal processor의 구성

Digital 신호처리 algorithm을 효율적으로, 빠른 속도를 수행할 수 있도록 특수하게 만들어진 기기를 Digital Signal Processor라고 한다.

새로운 신호처리 algorithm의 개발과 반도체 기술의 발전으로 보다 빠르고 성능이 우수한 각종 I.C.들이 개발됨에 따라, 보다 효율적이고 실시간 처리가 가능한 신호처리를 processor들이 개발되어 왔다.

신호처리 algorithm이 범용 computer에서 software에 의하여 처리되는 것과는 달리 microprogram memory에 각 instruction이 요구하는 control signal을 직접 저장하고 발생케 함으로써 더욱 빠른 계산속도를 얻을 수 있다.

(1) 구성 방법

고정된 명령어와 구조를 갖는 microprocessor는 일반적으로 사용하기에는 편리하지만 어떤 특수한 목적에 비추어 볼때 비효율적이고, 느리고, 많은 제약을 갖게 된다. 반면에 SSI/MSI소자들에 의한 구성은 System이 지나치게 복잡해지고 융통성이 없다는 단점을 갖는다⁵⁾.

따라서 programmable signal processor의 설계에는 word length의 결정이 용이하고, 필요한 명령어를 microprogram에 의해 최적으로 구성함으로써, 효율 및 속도를 증가시킬 수 있고 SSI/MSI design의 단점을 보

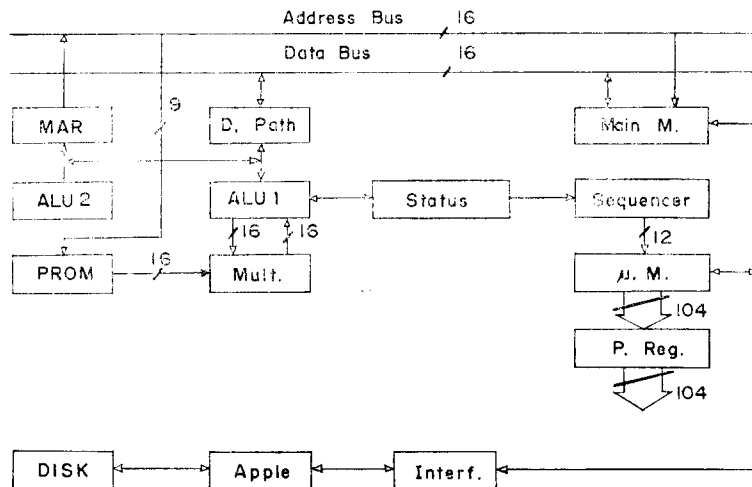


그림 1. 시스템의 구성도
Fig. 1. System block diagram

완할 수 있는 microprogrammable bit slice device에 의한 구성이 가장 적당하다고 생각된다.

본 processor는 Am2900 family bit-slice IC를 사용하여 pipelining을 고려한 구조를 갖고, horizontal microprogram에 의해 parallelism을 충분히 살린 신호 처리용 고속 processor이다. Control unit내의 microprogram memory는 writable control store(WCS)로 구성하였으므로 control unit에 내장되는 microprogram의 내용에 따라, 여러가지 digital 신호처리 algorithm을 실시간으로 수행할 수 있다. 본 processor는 특히 FFT algorithm의 실현에 알맞게 설계하였으나 digital filter 등으로의 응용도 가능하게 하였다.

(2) 구조

본 processor는 그림1과 같이 크게 ALU1과 ALU2의 dual ALU구조를 갖고 processor control unit(PCU)와 data path에 관계된 register block, main memory, clock 및 timing circuit, host computer와 interface의 부분으로 나누어질 수 있다.

ALU1은 4bit ALU slice Am2903A 4개와 high speed look-ahead carry generator Am2902A 1개, status and shift control unit Am2904 1개로 구성되며, 곱셈의 빠른 수행을 위해 16×16bit multiplier Am29517 1개가 연결되어 data처리에 필요한 덧셈, 곱셈 등의 연산을 담당한다. Am29517 multiplier는 FFT계산에 필요한 W^N 의 상수를 저장하고 있는 PROM과 Am2903A로부터 받은 data를 곱하여 그 결과를 다시 Am2903A로 보내준다. 여기에서 PROM에는 sine함수의 주기를 512points로 sampling하여 저장함으로써 2048 points까지의 FFT계산이 가능함을 알 수 있다.

ALU2는 4bit ALU slice Am2901 4개와 Am2902A 1개로 구성되어 있다. ALU2의 출력은 memory address register(MAR)에 연결되어서 memory 및 상수 저장용 PROM으로 필요한 address를 계산하여 보내주는 역할을 한다. ALU2는 74LS245 transceiver에 의하여 ALU1과의 data교환이 가능하다. 또한 ALU1이 butterfly계산을 하는 동안, 다음 계산에 필요한 data의 address를 계산하게 되므로 전체적인 계산속도가 증가하게 된다.

Data path는 수개의 register 및 process status word(PSW)의 처리에 필요한 회로로 구성되며, data bus로부터 ALU로 data를 보내주며, buffer의 역할을 한다. Am2904에 저장되는 status flag들은 직접 ALU1의 Y bus에 연결되기도 하고 PSW처리용 회로에 의해 ALU1의 DB bus로 연결되기도 하는데, 이 PSW은

test condition check 회로에 의해 하나가 선택되어져서 microprogram sequencer의 condition branch 등의 명령수행에 필요한condition test신호로 작동하게 된다.

Processor control unit(PCU)는 Am2910-1 microprogram sequencer에 의하여 4Kwords까지의 microprogram memory를 address 할 수 있고 16가지의 명령에 의하여 jump, conditional branch, subroutine, loop등의 다양한 기능을 수행할 수 있다. Microprogram memory는 microprogram의 수정이 용이하고, 여러가지 신호처리 algorithm들을 하나의 hardware에 의해 수행할 수 있도록 Intel 2148 static RAM에 의해 구성되었으며 host computer APPLE-II와 연결되어져서 microprogram을 받아들일 수 있도록 되어져 있다.

Main memory는 처리해야 할 data를 저장하고 또 처리된 결과가 저장되는 곳으로서 역시 host computer와 연결되어져 data의 전송이 가능하도록 하였다.

Host computer와의 interface는 104 bits의 microprogram instruction 및 16bit의 data length를 제어할 수 있도록 회로가 구성되었으며, 6502 assembler에 의하여 control program이 작성되었다.

(3) Microinstruction

본 processor는 microprogram에 의해 control 되는 resource들이 독립적인 unit로 분할 되어 있어서, 하나의 microinstruction이 동시에 모든 resource를 activate할 수 있는 horizontal microprogram을 사용하여 parallelism을 충분히 살렸다. 104bit로 구성되는 하나의 microinstruction은 그림2와 같이 크게 7개의 field로 나누어 진다.

103	93		90	78
Status (Am 2904)	R/W	CC EN	ALU 1 (Am 2903A)	
77	68 67	63 62	55 54	53
Datapath	Shift (Am 2904)	Register (Am 2903A)	Multiplier (Am 29517)	
51	44 43			26
Datapath	ALU2 (Am 2901)			
25	20 19	16 15	0	
Test0-Test5 (Am 2904)	Sequencer (Am 2910-1)	Immediate/Branch Address (Datapath/Am2910-1)		

그림 2. Microinstruction
Fig. 2. Microinstruction

(4) Microprogramming

본 연구에서는 microassembler를 사용하지 않고, 앞에서 정의한 microinstruction에 형식을 맞춘 coding sheet를 이용하여 microprogram을 작성하였다. 이런 방법으로 coding된 microprogram은 Micro-Tool 이라는 BASIC 및 6502 assembler로 작성된 프로그램에 의해 처리된다. Micro-Tool은 microprogram의 작성, 수정 및 확인, disk에의 저장 및 microprogram memory에의 전송등의 다양한 기능을 갖고 있다.

4. 결과 및 검토

Radix-2 DIT algorithm을 본 processor에서 실행 시키기 위하여 그림 3과 같은 flow chart에 의하여 별

첨1과 같은 microprogram을 작성하여 실행한 결과 다음과 같은 결과를 얻었다.

① APPLE-II와 signal processor 사이의 data transfer는 APPLE-II의 $\phi 1$ clock으로 30 cycles이 필요함으로 다음과 같음을 알 수 있다.

$$1 \text{ byte}/30 \times 10^{-6} \text{ sec} = 33 \text{ kbyte/sec}$$

②본 processor를 사용하여 FFT를 수행할 때 표1과 같이 하나의 butterfly계산에 약 34 cycles이 필요하다. 그러므로 1 MHz의 master clock을 사용하여 1024 points complex FFT를 할 경우 수행시간은 다음과 같다.

$$\frac{N}{2} \cdot \log_2 N \times 34 \text{ cycles} \times 10^{-6} \text{ sec}$$

$$(N=1024 \text{ 일 때 약 } 174\text{msec})$$

③ 본 processor는 look-up table의 용량에 의하여

별첨 1. FFT 프로그램 리스트
Appendix 1. Microprogram List of FFT

ADDR	MICROCODE
0000	00001000000000011111100010000111001001001010101010111011110001010100100011100110101111111
0001	00000000000010011111100010000101000100100111111010101011111111100101010010091100110101111111
0002	000000100000000111111000111000111011001011011110101011111110011110001010100100100110001111111
0003	00001000000000011111100011100011101100100111111010101011111011110101010101100110011001111111
0004	0000000000000001111110001001010101001001100111010001001000101011000001100100001111111
0005	00000000000001000111100011100011101100100010110101011110110111100010101000110110001101101
0006	000000000000010100111000111000111101100100101011010101111110111110101010100111001001111010
0007	00000000000001001111100011100011101100100100110010101011111101111110101010100111001001111101
0008	0000000000000110111110011100011101100100100111010101011111101111110101010100111001001111101
0009	0000101000000001111100011100011101100100000011011000100110111110010101001000110010001111111
0010	0000000000000001111110001000000000100100010111010111111011111001010101110010011010111111
0011	000000000000000110111000100000000011001000010111010110010001001110010101000000110010000111110
0012	000000000000000101011000010001110110010000011110101111110111110101010100111001001111110
0013	0000000000000001111110001000000001010010000111110101111110111110101010100111001001111111
0014	000000000000000111111000100001110110100011111101011111101111110101010100111001001111111
0015	0000000000000001111110000000001010100100010011101011111101111110101010100111001001111111
0016	0000000000000001111100001000101010100100010111101011111101111110101010100111001001111111
0017	11111110000000100001110000100100000010010011110101011111110011110101010010010010101010101
0018	000111000000000110001100010010101110011010111111101011111000111110101010100111001001111111
0019	00000000000001000111101010100001110110010010001110101011111100111110101010100111001001111111
0020	01010001000000000100010001000000011101001001001111010111111010111110101010100111001001111111
0021	00000000000000011111100010000100100100100111111010111111111111101010101001110010011111111
0022	000000000000000111111000100001110011010011111010101011110011100000101010001110010001111111
0023	0001101000000001110001000100101011110011010011110110100001110101101000101010000110010001111111
0024	0000000000000001111110000100001110011010011111110100001001011101010101001000110010001111111
0025	001011100000000111110000010000111001101001111101010011111001110010101010100111001001111111
0026	0010100000000001110011001000100000010010011110110100111111111010010101001000110010001010101
0027	0010100000000001110010000100001110011010011111110100010101011110101010101001001110100001111111
0028	1111111000000110111110001000011010011001001000111010101111110101111101010101000001001001001001
0029	0010000000000001100010010001010001100010010010011110101111110101111101010101001001001010101
0030	0101000100000000000100011000111010010010001110101111110101111101010101001110010011111111
0031	0051010100000000010010010010000111010010010011110101111110101111101010101001110010011111111
0032	1111111000000101000011000100001110101001001000111010101111110101111101010101000001001001001001
0033	01010100000000011000110010001010001001001001111010111110101111101010101001110010011111111
0034	0101000100000000000100010100000101010010010001111010111111010111110101010100111001001111111
0035	000101010000000001000100010000010001000100011110101111110101111101010101001110010011111111
0036	0000000000001000111100110110001110100100000110101011111101011111101010101001110010011111111
0037	01010001000000011100010001000011100110100111111010111111010111110101010100100111001001111111
0038	00000000000000000001110101010000111010010010011110101011111010111110101010100111001001111111
0039	000101010000000111001000010000111001101001111111010111110101111101010101001110010011111111
0040	00000000000000011111100001000011100110101111111010111111010101010011100100111111111111
0041	000000000000000111110001000000000001001010111111010111111111010010101001001100101011111111
0042	000000000000000111111000100001110011010111111101011111100011111010101010011100100111111111
0043	00110010000000000011000100010001001011100110101111110100011110101110001010100100111001001111111
0044	00000000000000011111000010000111001101001110011101010011101111010010101000001101100001111111
0045	000000000000000111111000010000111001101001110011010001000010111100010100000011011000001111111
0046	000000000000000111110000100001110011010011100111010100011011110000010101000001101100001111111
0047	0000000000000001111100001000011100110100111001110100000100101111010010101000001101100001111111
0048	0000000000000001111110000100001110011010011100111010010010101111010010101000001110100001111111
0049	000000000000000111110000100001110011010011101110101000110101110100101010000011011000001111111
0050	000000000000000111111000100000000001001001111111010101001111110010101001110010010011111111

표 1. FFT 도프로그램의 실행시간
Table 1. Execution time of fft
(1024 Point Complex FFT)

단 위	Clock수
Normalization	3
cos/sin Address 계산	4
Re X _{m+1} (p)	5.5
Re X _{m+1} (q)	4.5
Im X _{m+1} (p)	6.5
Im X _{m+1} (q)	4.5
Butterfly 마무리	6
계	34

1MHz의 경우 :

$$10 \times (N/2) \log_2 N \times 34 = 174 \text{ msec.}$$

2048 point까지의 FFT계산이 가능하다.

본 processor에서 1MHz의 master clock을 사용할 경우 1024 points complex FFT의 수행시간은 74m

sec로써 10MHz의 master clock을 사용하면 20msec이 내에 가능하다. 그리고 더욱 빠른 계산속도를 얻기 위해서는 hardware에서 ALU의 증가와 software에서 parallelism을 더욱 살림으로써 가능하다. 그리고 microprogramming은 coding sheet 및 Micro-Tool을 사용하여 수행했으나, 이 과정은 시간이 오래 걸리고 hardware의 구조에 대한 완벽한 이해가 있어야만 효율적인 microprogram을 작성할 수 있다는 문제가 있다. 그러므로 본 processor의 이용을 용이하게 하기 위해서는 본 processor 전용 microassembler 및 debuging program의 개발이 필요하다.

5. 결 론

본 16bit microprogrammable signal processor는 하나의 microinstruction이 104bit로 구성되는 horizontal microgram을 사용하여 processor 내부의 모든 부분이 microcode에 의하여 직접 control 되도록 함으로써

ADDR	MICROCODE
0051 :	000000000000001111110000100011110011010011111111010111111011111101010101001111001100111111111
0052 :	001101100000000011100110001000011110011010011111111100101111101111011101011001000011001000010111111
0053 :	01011010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0054 :	00000000000000011111100001000011110011010011111110010110101100011100101010100111011101010001111111
0055 :	0011100100000000011100110000100010100100100000001111100101111111111111011101011001000011001000010111111
0056 :	01011010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0057 :	0000000000000001111110000100001110011010011111110010001110100111011010101001100110001000011111111
0058 :	000000000000000111111000010000111001101001111111101011111101011111101011111101011111010011111111
0059 :	000000000000000111111000010000111001101001111111101011111111010111111010111110111001100111111111
0060 :	0011110000000000111001100100000001001001001111111110011000011111110111010110010000110010000101111111
0061 :	01011010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0062 :	00000000000000011111100001000011100110100111111100100011010001110110101010011101110101000011111111
0063 :	010000010000000011100110001000101001001000000111110011000111111011101011001000011001000010111111
0064 :	01011010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0065 :	00000000000000011111100001000011100110100111111100100011111100100011111100101011001000010001010111001
0066 :	01001010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0067 :	000000000000000111111000010000110011001000111111101010111101111010101010011000100110000101111111
0068 :	01001010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0069 :	000000000000000111111000010001110011001000010111101000110110100011011001001110010001110010001111111
0070 :	00000000000000011111100001000111001100100001011111010011011011111001010100100011100100011111111
0071 :	0000000000000001111110001000010011001000000100110010000001110101111110111111010101010001110010001111111
0072 :	0000000000000001111110001000000100110010000011111010101111011111010101010011000110110001101111111
0073 :	0000111000000000100011000100100111110011010011111111010011101001111110101011101000011001000010111111
0074 :	00000000000000011111100010000110111001001110011101011101000001100010101010000111001000011111111
0075 :	00000000000000011001110001000001011110010010101111010101010000011100101010000001100100001111110
0076 :	01000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0077 :	00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0078 :	00001001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0079 :	00001001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0080 :	00000000000000011111100010000111001100100111111101011111110101111111010111111010101100111001001111110
0081 :	00000000000000011111100010000111001100100111111110101111111101011111110101111111010101100111001001111111
0082 :	000000000000000111111000100001110011001001111111101011111111010111111110101111111010101100111001001111111
0083 :	00000000000000011111100010000111001100100111111110101111111101011111111101011111111010101100111001001111111
0084 :	00000000000000011111100010000111001100100111111110101111111101011111111101011100100011101100011111111
0085 :	00000000000000011111100010000111001100100111111110101111111101011111110101111110101011001100010001100110001111111
0086 :	0000000000000001111110001000011000010000010001011111110101011111111101010111111110000101010001001100110001111111
0087 :	000000000000000111111000100001110011001001111111010101111111010101111111000110101100011001100011111111
0088 :	000000000000000111001100010001001011100110100111101110101011110111101001010110010001100100010101111111
0089 :	00000000000000011110001000011100110010011111110101111111010101111011010101100100011101100011111111
0090 :	01011100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0091 :	0000000000000001111110001000011100110010011111111010111111110101111111101011111110101011001100010001100110001111111
0092 :	000000000000000111111000100001110011001001111111101011111111010111111111010111111110101011001100010001100110001111111
0093 :	000000000000000111111000100001110011001001111111101011111111010111111111010111111110101011001100010001100110001111111
0094 :	00000000000000011111100010000111001100100111111110101111111101011111111101011100100011101100011101100011111111
0095 :	0000000000000001111110001000011100110010011111111010111111110101111111110101011001100011100100011100100011111111
0096 :	00000000000000011111100010000111001100100111111110101111111101011111111101011111110101111110100011011111111

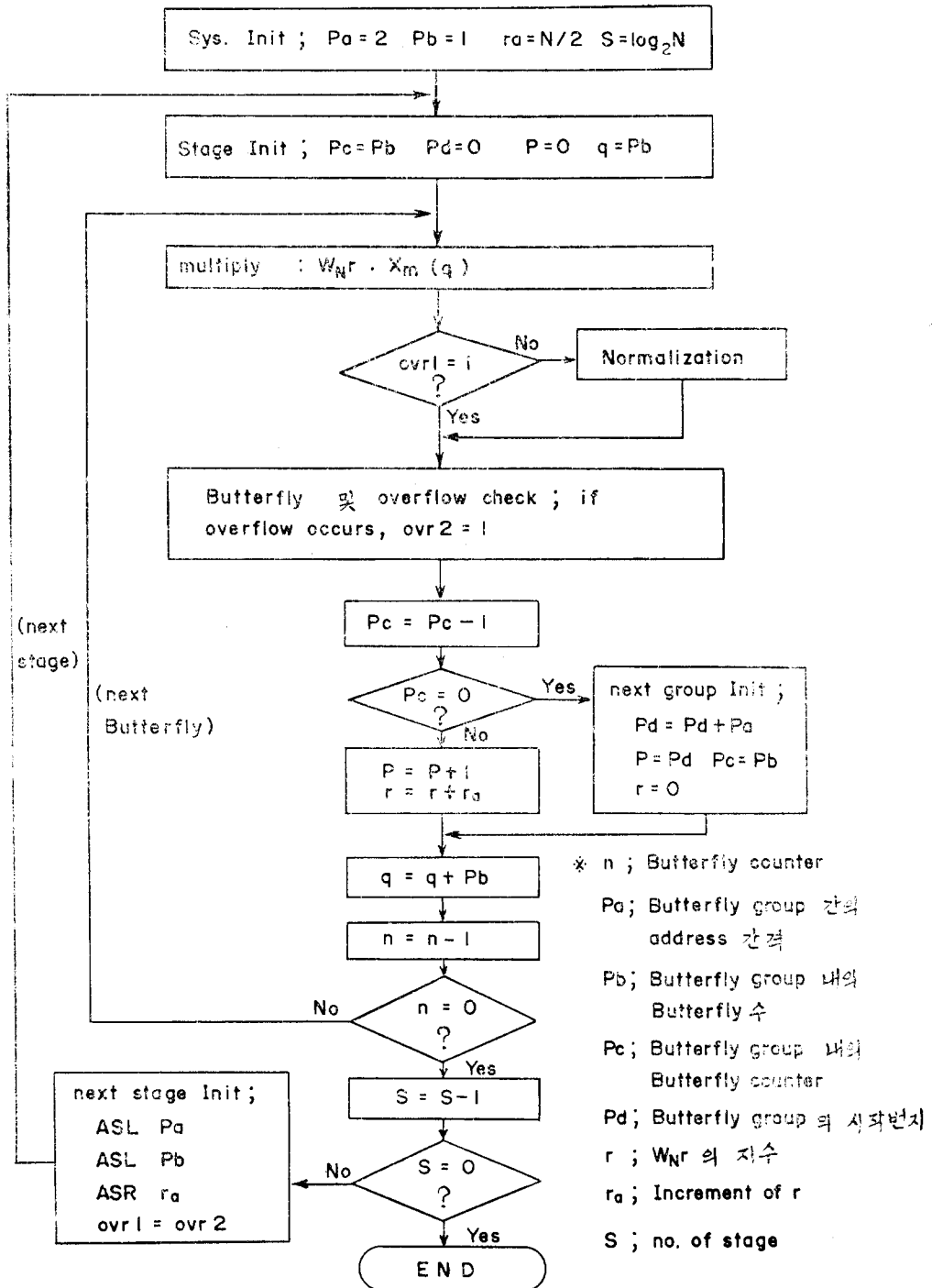


그림 3. FFT 처리흐름도
 Fig. 3. Flowchart for FFT

control signal이 decoding되는 과정이 불필요하고 이로 인해 계산속도의 증가를 얻을 수 있었다. 또한 pipelined architecture을 갖도록 설계하였으므로 data dependency를 따져서 pipelining을 실현시킴으로써 계산속도를 더욱 증가시킬 수 있음을 확인할 수 있었다.

Processor control unit는 writable control store (WCS)로 구성하고 host computer와 interface하였으며 microprogram에 의해 각종 digital 신호처리 algorithm을 수행할 수 있도록 하였다.

(상기의 연구는 한국학술진흥재단 '84첨단과학 연구비에 의하여 이루어진 것으로 본지를 빌어서 감사를 드린다).

참 고 문 헌

- 1) L.R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing," Prentice-Hall, 1975, pp.356~482.
- 2) A.C. Salazar, "Digital Signal Computers & Processors," IEEE Press, 1977.
- 3) R. Kuc and M. Schwartz, "Estimating the Acoustic Attenuation Coefficient Slope for Liver from Reflected Ultrasound Signals," IEEE Sonics and Ultrasonics, Vol. 26~5, pp. 353~362, 1979.
- 4) E.A. Robinson, "A Historical Perspective of Spectrum Estimation," IEEE Proc. Vol. 70, No.9 Sep. 1982.
- 5) J. Mick and J. Brick, "Bit-Slice Microprocessor Design," McGraw-Hill, 1980.