

디지털 필터 전용처리기의 설계와 구성에 관한 연구

84308

○남상엽*, 박인정*, 이태원**

* 단국대학교

**고려대학교

A Study ON Design and Construction of a Special Processor for Digital Filtering

○Sang Yep Nam*, In Jung Park*, Tae Won Rhee**

* Department of Electronic Eng., Dankook Univ., **Korea Univ.

Abstract

A microprocessor for the dedicated purpose of digital filtering has been designed and constructed. A microsequencer to control the flowing of data is designed with the concept of a hardware compiler of AHPL statements. In order to save the processing time, this system utilizes parallel data calculation method instead of the one proposed by Peled and Liu which depended on serial calculation method. The flexible program control technique has been applied to the implementation of the special processor for the digital signal processing.

1. 서 론

규모가 큰 디지털 시스템을 부울함수나 상태표로 표시하지 않고 Subsystem의 합수에 의한 표현을 하기 위하여 A.H.P.L. (A Hardware Programming Language,)로써 표시할 수 있다.⁽¹⁾

A.H.P.L. 표시에서는 레지스터를 디지털 시스템의 기본적인 소자들로 쓴다. 이런 표시 방법을 사용함으로써 레지스터에 있는 데이터들을 간략하고 정확하게 어떻게 움직이며 또한 어떤 프로세싱을 하느냐 하는 것을 표시할 수 있다.

이러한 설계방법을 다음과 같은 순서로 표시한다.⁽²⁾

- (a) 구조정의 (Architecture)
- (b) 기체조직 (Machine Organization)
- (c) 제어부분 (Control)
- (d) 데이터부분 (Data Flow)

본 연구에서는 특수목적의 신호처리 프로세서를 A.H.P.L.로 설계하고 구성하여 디지털 필터링을

실현하고자 한다.

2. 프로세서에 적용한 이론

입력 데이터 어드레싱 방법에 ⁽⁴⁾ 기초를 두고 프로그램을 설계했다.

이때 입출력관계는 다음과 같이 표시된다.

$$y(n) = [ROM]^{1 \times n} + [ROM]^{1 \times (n-1) + offset_1} \\ + [ROM]^{1 \times (n-2) + offset_2} + \dots + [ROM]^{1 \times (n-1) + offset_1} \\ - [ROM]^{1 \times (n-1) + offset_1} - [ROM]^{1 \times (n-2) + offset_2} \\ \dots$$

X(n)이 00부터 FF₁₆까지 변화할 때의 각 계수와 X(n)의 꼽을 위해 다음과 같이 메모리를 할당해 주었다.

K	계수·시퀀스	실효어드레스범위
0	a(0) · X(n)	000 ~ 0 FF
1	a(1) · X(n-1)	100 ~ 1 FF
2	a(2) · X(n-2)	200 ~ 2 FF
3	a(3) · X(n-3)	300 ~ 3 FF
4	b(1) · Y(n-1)	400 ~ 4 FF
5	b(2) · Y(n-2)	500 ~ 5 FF
6	b(3) · Y(n-3)	600 ~ 6 FF
7	b(4) · Y(n-4)	700 ~ 7 FF

곱셈 테이블 작성을 위해 승산 프로그램을 사용하고 ROM WRITER에 의해 이를 승산값을 ROM에 저장한다.

사용자 메모리는 800 ~ 8FF 까지 쓴다.

3. 프로세서의 설계

일반적인 I.I.R. 디지털 필터링 하드웨어의 Block Diagram은 그림 1과 같이 표시되고 다음과 같은 부분으로 구성되어 있다.⁽⁶⁾

- (a) 입력샘플 및 출력샘플을 위한 레지스터
- (b) 계수 승산치 메모리
- (c) 승산 및 합산을 위한 A.L.U.
- (d) 동작순서 제어를 위한 제어부

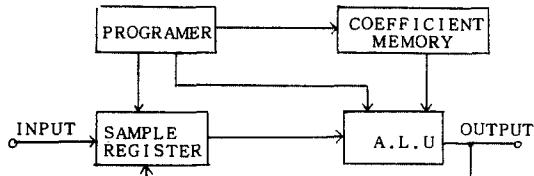


그림 1. 전형적인 디지털 필터링 하드웨어 프로세서

(1) 명령어의 종류 및 마이크로 시퀀스

본 연구에서 구성하는 신호처리전용 마이크로 프로세서에 사용된 4비트의 명령어를 마이크로 명령어의 개념에 입각하여 다음 표 1에서와 같이 재정하였고 본 디지털 시스템의 제어부분 설계를 하는데 제어 순서(Control Sequence)를 다음 표 2에서 같이 A.H.P.L.로 쓸 수 있다.

(표 1) 명령어 세트

IR ₃	IR ₂	IR ₁	IR ₀	MNEMONIC
0	0	0	0	LOAD SX ₀ , X(n)
0	0	0	1	LOAD SX ₁ , X(n-1)
0	0	1	0	LOAD SX ₂ , X(n-2)
0	0	1	1	LOAD SX ₃ , X(n-3)
0	1	0	0	LOAD SY ₁ , Y(n-1)
0	1	0	1	LOAD SY ₂ , Y(n-2)
0	1	1	0	LOAD SY ₃ , Y(n-3)
0	1	1	1	LOAD SY ₄ , Y(n-4)
1	0	0	0	LOAD X(n), A/D
1	0	0	1	COMPUTE Y(n)
1	0	1	0	LOAD Y(n), F
1	0	1	1	OUTPUT Y(n)
1	1	0	0	SHIFT DOWN X _i
1	1	0	1	SHIFT DOWN Y _i
1	1	1	0	SHIFT LEFT Y(n)
1	1	1	1	SHIFT RIGHT Y(n)

(표 2) 마이크로 시퀀스

MODULE: SPECIAL-PURPOSE MICRO PROCESSOR
FOR DIGITAL FILTERING

MEMORY: M[4096,8]; X[8]; Y[8]; SX[8]; SY[8];
C[12]; D[12]; F[12]; MA[12]; PC[12];
MD[8]; IR[4]

1. → (START, START)/(1, 2)
2. PC ← 8φφ
3. MA ← PC
4. MD ← M^{MA}, PC ← PC + 1
5. IR ← MD
6. → (IR₃, IR₃)/(7, 26)
7. → (IR₂, IR₂)/(8, 17)
8. → (9, 11, 13, 15) ⊥ (IR₁, IR₀)
9. OUTPORTφ ← X(n)
10. SX₀ ← M^{outportφ}, → (3)
11. OUTPORT1 ← X(n-1)
12. SX₁ ← M^{outport1}, → (3)
13. OUTPORT2 ← X(n-2)
14. SX₂ ← M^{outport2}, → (3)
15. OUTPORT3 ← X(n-3)
16. SX₃ ← M^{outport3}, → (3)
17. → (18, 20, 22, 24) ⊥ (IR₁, IR₀)
18. OUTPORT4 ← Y(n-1)
19. SY₁ ← M^{outport4}, → (3)
20. OUTPORT5 ← Y(n-2)
21. SY₂ ← M^{outport5}, → (3)
22. OUTPORT6 ← Y(n-3)
23. SY₃ ← M^{outport6}, → (3)
24. OUTPORT7 ← Y(n-4)
25. SY₄ ← M^{outport7}, → (3)
26. → (IR₂, IR₂)/(27, 32)
27. → (28, 29, 30, 31) ⊥ (IR₁, IR₀)
28. X(n) ← A/D, → (3)
29. C₁ ← SX₀ + SX₁, C₂ ← SX₂ + SX₃,
C₃ ← SY₁ + SY₂, C₄ ← SY₃ + SY₄,
D₁ ← C₁ + C₂, D₂ ← C₃ + C₄,
F ← D₁ + D₂, → (3)
30. Y(n) ← F, → (3)
31. D/A ← Y(n), → (1)
32. → (33, 36, 40, 41) ⊥ (IR₁, IR₀)
33. X(n-3) ← X(n-2)
34. X(n-2) ← X(n-1)

35. $X(n-1) \leftarrow X(n)$
 $\rightarrow (3)$
 36. $Y(n-4) \leftarrow Y(n-3)$
 37. $Y(n-3) \leftarrow Y(n-2)$
 38. $Y(n-2) \leftarrow Y(n-1)$
 39. $Y(n-1) \leftarrow Y(n), \rightarrow (3)$
 40. $Y(n) \leftarrow SL(Y(n)), \rightarrow (3)$
 41. $Y(n) \leftarrow SR(Y(n)), \rightarrow (3)$

(2) Data Path 및 Control Sequence 설계

Data Path 설계는 Destination and Source List에서 Destination Group이 있는 것에 Source에 있는 메이타가 Clock과 Source 조건에 의하여 전송된다. 이와 같은 방법을 병렬 메이타 어드레싱에 의하고 병렬 연산방식의 기본적 개념으로(그림 2)와 같이 설계하였다.

Control Sequencer는 마이크로 시퀀스에서 메이타를 레지스터나 연산기로 전송할 때 이를 조정하는 Control table에서 제어 순서가 결정되어 실행하는 논리시스템을 Control Sequencer라 하며 각 Micro-operation을 실행하는데 필요한 Timing pulse를 출력으로 낸다. 이 제어 시스템 설계는 한 State에 대해서 한 Flip-flop 방법을 썼다.

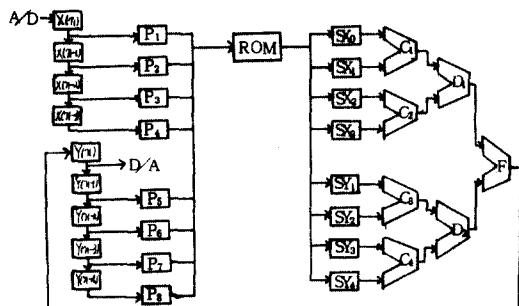


그림 2 Data Path

4. 실험

본 연구에서 디지털 필터링을 위한 AHPL 프로세서를 구성하여 실험하기 위하여 구성한 하드웨어 구성을 그림 3과 같다.

하드웨어 구성화로는 (1)샘플홀드회로 (2)A/D 변환회로 (3)AHPL 신호처리전용 마이크로 프로세서 (4)D/A

변환회로 (5)전원 공급회로이다.



그림 3. 하드웨어 구성도

5. 결론

대규모 시퀀스 시스템의 설계는 새로운 방법으로 추구되어야 하는바 그 방법으로 컴퓨터의 합성 실현 방법인 Transfer logic 즉 AHPL로써 시스템을 구성하였다.

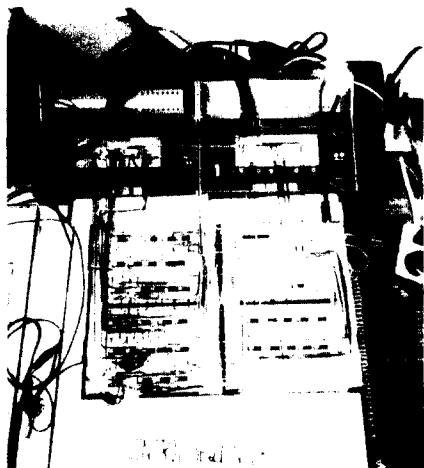
AHPL 방법은 어떤 특정 목적으로 설계하기가 편리 하므로 이 방법을 썼고, 메이타 직렬비트 입력에 의하여 ROM의 어드레스를 지정하는 Peled, Liu⁽²⁾ 메이타 연산 방식보다 본 논문에서 실현한 병렬 메이타 어드레싱에 의한 병렬 연산방식이 메이타 처리 속도를 향상시키고 Peled, Liu 병렬 메이타 계산 방식 보다는 융통성이 있게 프로그램 제어 방식을 사용했다. Peled, Liu의 경우 ROM의 값은 하나하나 계산하여 집어 넣어야 하지만 본 논문에서 적용한 방법에서는 컴퓨터 프로그램에 의해 즉시 계산하여 저장하는 것이 가능하다.

앞으로의 과제는 여러 필터 함수를 적용하고 data path를 보강하여 일반화 하면 될 것이다.

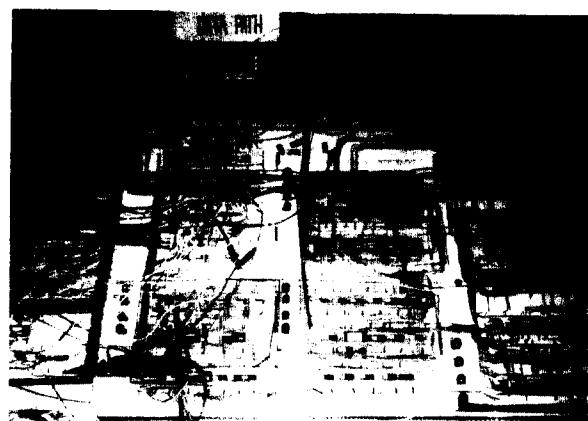
※ 참고문헌

1. FREDRICK.J.HILL AND GERALD.R.PETERSON: DIGITAL SYSTEMS; HARDWARE ORGANIZATION AND DESIGN, JOHN WILEY & SONS, 1978.
2. A PELED AND B.LIU: DIGITAL SIGNAL PROCESSING, JOHN WILEY AND SONS, 1976.
3. M. MORRIS MANO: DIGITAL LOGIC AND COMPUTER DESIGN, PRENTICE-HALL, 1979.
4. 이태원, 박인정: "Z-8φ에 의한 디지털 필터 실현에 관한 연구", 대한전자공학회 하계종합 학술대회 논문집, VOL.6, NO.1, 1983.7.
5. V.CAPPELLINI: "DIGITAL FILTERS AND THEIR APPLICATIONS", A CADEMIO PRESS, Chap.7, 1978.

부 록



Control Sequencer 실험장치



Data Path 실험장치