

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

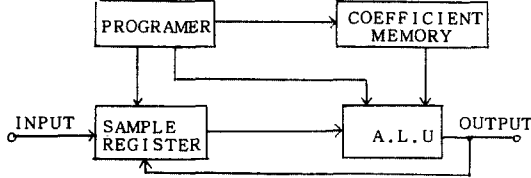


그림 1. 전형적인 디지털 필터링 하드웨어 프로세서

(1) 명령어의 종류 및 마이크로 시퀀스

본 연구에서 구성하는 신호처리전용 마이크로 프로세서에 사용된 4비트의 명령어를 마이크로 명령어의 개념에 입각하여 다음 표 1에서와 같이 제정하였고 본 디지털 시스템의 제어부분 설계를 하는데 제어 순서(Control Sepuencer)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, $\overline{\text{START}}$)/(1, 2)
2. PC ← 8φ
3. MA ← PC
4. MD ← M^{MA}, PC ← PC + 1
5. IR ← MD
6. → ($\overline{\text{IR}}_3$, IR₂)/(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. → ($\overline{\text{IR}}_2$, 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 List 에서 Destination Group 이 있는 것에 Source 에 있는 데이터가 Clock 과 Source 조건에 의하여 전송된다. 이와같은 방법을 병렬 데이터 어드레싱에 의하고 병렬 연산방식의 기본적 개념으로(그림 2)와 같이 설계하였다.

Control Sequencer 는 마이크로 시퀀스에서 데이터를 레지스터나 연산기로 전송할 때 이를 조정하는 Control table 에서 제어 순서가 결정되어 실행하는 논리시스템을 Control Sequencer 라 하며 각 Micro-operation 을 실행하는데 필요한 Timing pulse를 출력으로 낸다. 이 제어 시스템 설계는 한 State 에 대해서 한 Flip-flop 방법을 썼다.

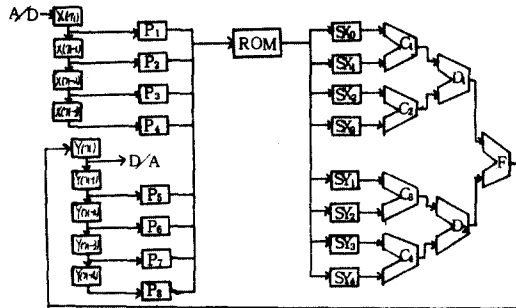


그림 2 Data Path

4. 실험

본 연구에서 디지털 필터링을 위한 AHPL 프로세서 구성하여 실험하기 위하여 구성된 하드웨어 구성도는 (그림 3)과 같다.

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

변환회로 (5)전원공급회로이다.



그림 3 하드웨어 구성도

5. 결론

대규모 시퀀스 시스템의 설계는 새로운 방법으로 추구되어야 하는데 그 방법으로 컴퓨터의 합성 실현 방법인 Transfer logic 즉 AHPL 로써 시스템을 구성하였다.

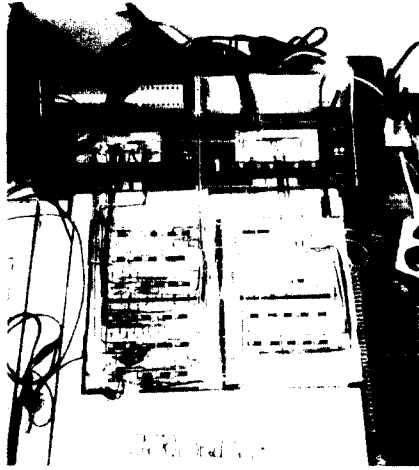
AHPL 방법은 어떤 특정 목적으로 설계하기가 편리하므로 이 방법을 썼고, 데이터 직렬비트 입력에 의하여 ROM 의 어드레스를 지정하는 Peled, Liu⁽²⁾ 데이터 연산 방식보다 본 논문에서 실현한 병렬 데이터 어드레싱에 의한 병렬 연산방식이 데이터 처리 속도를 향상시키고 Peled, Liu 병렬 데이터 제산 방식보다는 융통성이 있게 프로그램 제어 방식을 사용했다. Peled, Liu 의 경우 ROM 의 값은 하나하나 제산하여 집어 넣어야 하지만 본 논문에서 적용한 방법에서는 컴퓨터 프로그램에 의해 즉시 제산하여 저장하는 것이 가능하다.

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

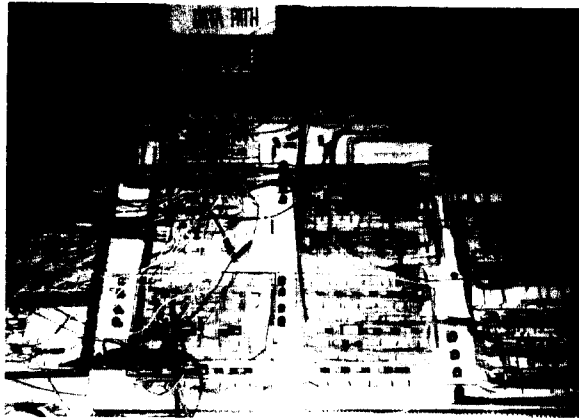
※ 참고문헌

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. CAPELLINI: "DIGITAL FILTERS AND THEIR APPLICATIONS", A CADEMIO PRESS, Chap.7, 1978.

부 록



Control Sequencer 실험장치



Data Path 실험장치