

工學博士 學位論文

증기터빈용 디지털 조속기의
설계에 관한 연구

A Study on the Design of a Digital Governor
for Steam Turbines

指導教授 金潤植

2007 年 2 月

韓國海洋大學校 大學院
電氣電子工學科 宋東永

목 차

목차	①
List of Figures	③
List of Tables	⑫
Abstract	⑭
1. 서론	1
2. 하드웨어의 설계	5
2.1 조속기 원리	5
2.2 조속기 운전 방법 및 특징	10
2.3 하드웨어 설계	17
3. 소프트웨어의 설계	55
3.1 개요	55
3.2 시스템 프로그램 구성	55
3.3 프로그램 개발 내용	57
4. 시스템 알고리즘의 구현	82
4.1 복합블록	82
4.2 단일블록	96
5. 속도제어 알고리즘의 동조와 시뮬레이션.....	106
5.1 제어대상의 모델링과 파라미터.....	106
5.2 조속기의 설계 알고리즘	115

5.3 시뮬레이션 및 고찰	122
6. 결론	129
참고문헌	131
부록	135
감사의 글	140

List of Figures

그림 2.1 기계식 조속기

Fig. 2.1 Mechanical governor

그림 2.2 볼 헤드 속도감응장치

Fig. 2.2 Equipment of speed sensing ball head

그림 2.3 어큐물레이터와 서보기구

Fig. 2.3 Accumulator type oil pressure pump and servomechanism

그림 2.4 단독부하 등시성 운전

Fig. 2.4 Isochronous mode at isolated load

그림 2.5 단독부하 스피드드롭운전

Fig. 2.5 Speed droop mode at isolated load

그림 2.6 동일 용량 동일 스피드드롭의 2기운전

Fig. 2.6 Same capacity and speed regulation (at 2 sets operation)

그림 2.7 스피드드롭이 다른 동일 용량2기운전

Fig. 2.7 Same capacity and different speed regulation(at 2 sets operation)

그림 2.8 디지털 거버너 개념도

Fig. 2.8 Configuration of the digital governor system

그림 2.9 시스템 계통도

Fig. 2.9 Block diagram of the system

그림 2.10 CPU 모듈 내부 구조

Fig. 2.10 Block diagram of CPU module

그림 2.11 어드레스 디코더 타이밍도

Fig. 2.11 Time chart of address decoder

그림 2.12 웨이트 제너레이터 타이밍도

Fig. 2.12 Time chart of wait generator

그림 2.13 SRAM 타이밍도

Fig. 2.13 Time chart of SRAM

그림 2.14 Low power SRAM 타이밍도

Fig. 2.14 Time chart of low power SRAM

그림 2.15 EPROM 타이밍도

Fig. 2.15 Time chart of EPROM

그림 2.16 리셋 타이밍도

Fig. 2.16 Time chart of RESET

그림 2.17 SIO 모듈 블록 다이어그램

Fig. 2.17 Block diagram of SIO module

그림 2.18 키패드 구성

Fig. 2.18 KEYPAD

그림 2.19 LCD회로 블록 다이어그램

Fig. 2.19 Block diagram of LCD

그림 2.20 WRITE 타이밍도

Fig. 2.20 Time chart of WRITE

그림 2.21 READ 타이밍도

Fig. 2.21 Time chart of READ

그림 2.22 터빈의 회전속도 측정

Fig. 2.22 Measurement of turbine RPM

그림 2.23 DSSM 블록 다이어그램

Fig. 2.23 Block diagram of DSSM

그림 2.24 카운터 초기화 타이밍 선도

Fig. 2.24 Time chart of counter for initializing

그림 2.25 스피드센서 PAL 측정파형

Fig. 2.25 Waveform of Speed sensor PAL

그림 2.26 2진 동기카운터

Fig. 2.26 Synchronous binary counter

그림 2.27 2진 동기카운터의 타이밍선도

Fig. 2.27 Time chart of synchronous binary counter
(Serial carry type)

그림 2.28 스피드센서 카운터회로 측정 파형

Fig. 2.28 Waveform of Speed sensor counter circuit

그림 2.29 아날로그 출력모듈 블록도

Fig. 2.29 Analog output module block diagram

그림 2.30 D/A 컨버터의 타이밍도

Fig. 2.30 Time chart of D/A converter

그림 3.1 시스템 프로그램 구성도

Fig. 3.1 Structure of system program

그림 3.2 프로그램 RATE GROUP TIMING

Fig. 3.2 Timing of program rate group

그림 3.3 CPU 테스트 프로그램 흐름도

Fig. 3.3 Flow chart of CPU test program

그림 3.4 데이터용 SRAM의 흐름도

Fig. 3.4 Flow chart of SRAM for data

그림 3.5 명령용 SRAM의 흐름도

Fig. 3.5 Flow chart of SRAM for instruction

그림 3.6 ROM 테스트의 흐름도

Fig. 3.6 Flow chart of ROM for test

그림 3.7 백업용 SRAM의 흐름도

Fig. 3.7 Flow chart of SRAM for backup

그림 3.8 RATE 타이머 흐름도

Fig. 3.8 Flow chart of SRAM for data

그림 3.9 리얼타임 클럭 흐름도

Fig. 3.8 Flow chart of real time clock

그림 3.10 시리얼 커뮤니케이션 컨트롤러 흐름도

Fig. 3.10 Flow chart of serial communication controller

그림 3.11 부트스트랩 흐름도

Fig. 3.11 Flow chart of bootstrap

그림 3.12 타이머 초기화 흐름도

Fig. 3.12 Flow chart of timer initializing

그림 3.13 리얼타임 초기화 흐름도

Fig. 3.13 Flow chart of real time clock initializing

그림 3.14 시리얼 커뮤니케이션 컨트롤러 초기화 흐름도

Fig. 3.14 Flow chart of serial communication controller initializing

그림 3.15 파워 다운 흐름도

Fig. 3.15 Flow chart of power down

그림 3.16 스피드센싱 흐름도

Fig. 3.16 Flow chart of speed sensing

그림 3.17 타임스케줄 흐름도

Fig. 3.17 Flow chart of time schedule

그림 3.18 SCC 리시버 흐름도

Fig. 3.18 Flow chart of SCC receiver

그림 3.19 키패드 흐름도

Fig. 3.19 Flow chart of keypad

그림 3.20 RS232 흐름도

Fig. 3.20 Flow chart of RS232

그림 3.21 LCD 컨트롤러 흐름도

Fig. 3.21 Flow chart of LCD controller

그림 3.22 속도 변환 흐름도

Fig. 3.22 Flow chart of speed regulation

그림 3.23 인터 커뮤니케이션과 보팅 개념도

Fig. 3.23 Flow chart of inter communication and voting

그림 3.24 인터 커뮤니케이션 흐름도

Fig. 3.24 Flow chart of inter communication

그림 3.25 보팅 흐름도

Fig. 3.25 Flow chart of voting

그림.4.1 MPU 블록

Fig. 4.1 MPU_BLOCK

그림 4.2 스피드센서 블록

Fig. 4.2 SPD_SENSOR_BLOCK

그림 4.3 ACKNOWLEDGE_블록

Fig. 4.3 ACKNOWLEDGE_BLOCK

그림 4.4 밸브 드라이버 블록

Fig. 4.4 VAVL_DRVR_BLOCK

그림 4.5 액츄에이터 출력 블록

Fig. 4.5 ACT_OUT_BLOCK

그림 4.6 셧 다운 블록

Fig. 4.6 SHUTDOWN_BLOCK

그림 4.7 아날로그 입력 A 블록

Fig. 4.7 ANALOG_IN_A_BLOCK

그림 4.8 아날로그 입력 C 블록
Fig. 4.8 ANALOG_IN_C_BLOCK

그림 4.9 콘택 입력 로 블록
Fig. 4.9 CONTACT_INPUT_LOW_BLOCK

그림 4.10 콘택 입력 하이 블록
Fig. 4.10 CONTACT_INPUT_HIGH_BLOCK

그림 4.11 콘택 출력 블록
Fig. 4.11 CONTACT_OUT_BLOCK

그림 5.1 발전용 증기 터빈의 속도제어 시스템
Fig. 5.1 Speed control system of steam turbine for power plant

그림 5.2 제어밸브의 동작 특성
Fig. 5.2 Operating characteristic of control valve

그림 5.3 증기 vessel
Fig. 5.3 Steam vessel

그림 5.4 속도와 토크사이의 관계
Fig. 5.4 Relation of speed and torque

그림 5.5 제어대상의 블록선도
Fig. 5.5 Block diagram of control target

그림 5.6 하이브리드 유전알고리즘의 동작과정

Fig. 5.6 Process of H.G.A.

그림 5.7 제어대상 시스템의 파라미터 추정 구성도

Fig. 5.7 Diagram of parameter for control target

그림 5.8 속도/부하 제어 알고리즘을 갖는 제어계

Fig. 5.8 Control system of SPEED/LOAD algorithm

그림 5.9 PID 제어

Fig. 5.9 PID Control

그림 5.10 Relay 피드백 제어 시스템

Fig. 5.10 Control system of relay feedback

그림 5.11 스텝응답

Fig. 5.11 Step response

그림 5.12 스텝상의 외란을 가함

Fig. 5.12 Step response to the perturbation

그림 5.13 부하감쇄계수 변동시 응답특성

Fig. 5.13 Step response to the load-damping constant

그림 5.14 외란과 부하감쇄계수 변동을 고려한 응답특성

Fig. 5.14 Step response to the perturbation and load-damping constant

List of Tables

표 2.1 어드레스 맵

Table 2.1 Address map

표 2.2 리드/라이트 사이클에 의한 버스 사이즈

Table 2.2 BUS size for READ/WRITE cycle

표 2.3 인터럽트 ASSIGNMENT

Table 2.3 INTERRUPT ASSIGNMENT

표 2.4 SIO 모듈 어드레스 맵

Table. 2.4 SIO module address map

표 2.5 LCD 컨트롤러핀 기능

Table 2.5 Function of LCD CONTROLLER PIN

표 2.6 속도별 사이클타임

Table 2.6 Cycle time of turbine

표 2.7 로직선택과 어드레스맵

Table 2.7 Select logic and address map

표 2.8 아날로그 출력모듈 어드레스 선택

Table 2.8 Analog output module address selection

표 5.1 Ziegler-Nichols의 스텝응답을 이용한 동조

Table 5.1 Tuning for step response of Ziegler-Nichols

표 5.2 한계감도법에 의한 동조

Table 5.2 Tuning for marginal sensitivity

표 5.3 시뮬레이션 파라미터

Table 5.3 Parameters for simulation

A Study on the Design of a Digital Governor for Steam Turbines

by Song Dong-Young

Department of Electrical and Electronics Engineering
The Graduate School of Korea Maritime University
Busan, Republic of Korea

Abstract

In our modern societies, electric power is the driving force of the industry. As many parts rely on electric power, demand for electricity usually rises. In case of most Korean power plants, a mechanical or electronic governor is used for

an actuator control of a turbine generator.

For the stable power distribution service, it is desirable to change the old ones with improved governor of the steam turbine in speed control of the steam turbine at power plant for a long time.

In this paper, hardware and software system of a digital governor are designed to develop a controller of the digital type governor for the steam turbine.

Hardware in this digital governor consists of duplex I/O module and triplex CPU module and can drive a servo valve using 2 out of 3 voting algorithm with self-diagnostic ability.

Software in this digital governor consists of program of the operating system and applied programs.

The algorithm of speed control of the steam turbine is analyzed for the digital governor of Woodward Governor Company and made up for the weak points.

Transfer function is gained from model of a target plant and then combined into the overall transfer function. Then parameters are estimated, through an optimum object function.

The availability of a designed controller is confirmed through numerical simulation and desired results are attained.

제 1 장 서론

1.1 연구의 배경

현대 산업에서 전력은 산업의 원동력이라 할 수 있고 안정적인 발전 및 공급은 에너지 자원이 부족한 우리나라의 경우 풍요로운 문화생활과 더불어 경제발전을 이어가기 위해서는 늘어나는 전력수요에 맞추어 화력, 수력, 원자력 등 여러 가지 에너지원의 기술적 경제적 특성을 살린 최적의 운용으로 안정적인 전력을 확보하는 것이 주요한 정책과제로 되어 있다.

우리나라의 장기 전력 수급계획을 보면 1998년부터 2015년까지 향후 연평균 3.8[%] 수준으로 성장하여 2015년까지 6,957만[kW]까지 증가할 것으로 전망되는 전력수요의 증가에 대응하여 전력수요관리 및 기술개발 강화로 전력설비 증설을 억제하는 한편, 2015년까지 원자력 18기, 1,860만[kW]등 총 1백 17기 5,159만[kW]의 발전설비를 건설하여 발전설비를 8,083만[kW]으로 확충하고 송·변전설비를 적기에 건설하여 전력수급에 원활을 기한다는 것이다^[1].

따라서 본 논문에선 장기적으로는 안정되게 전력을 공급할 수 있는 전력공급기술을 위해 현재 발전소에서 다양하게 사용되고^[2] 있는 증기터빈의 조속기에 대한 성능개선을 통한 최적의 전력공급 및 설비의 안정을 기하고자 한다.

발전소에서 터빈 발전기의 액츄에이터를 제어하는 조속기는 구성방식에 따라 기계식 조속기, 기계 유압식 조속기 및 디지털 조속기 등 다양한 형태를 보여주고 있고^[3] 현존하는 국내 발전소 경우 대부분이 기계 유압식 및 전기전자식 조속기를 채용하고 있다^[4].

기계식조속기의 경우는 작동부의 마모 및 설비의 노후로 인해 정비 시 장시간의 경과로 인해 부품 공급 상의 문제를 초래할 수도 있다. 또한 전

자식의 경우는 기계식 조속기에 이러한 문제들이 다소간 경미 할 수 있으나 전자회로상의 부품의 열화시 문제가 대두될 수도 있다^[5].

이러한 문제점은 장기적인 측면에서 본다면 전력공급에 치명적인 차질을 초래할 수도 있다. 최근의 전기전자의 기술의 발달과 다양한 제어기법의 등장 디지털 기술의 비약적인 진보는 기기수명의 연장, 내구성의 증대, 현대제어기법의 적용을 통한 안정적 운용과 강인한 과도특성 및 사용의 편의 증진 등 다양하고 많은 장점을 제공하고 있다.

한편으로는 이러한 기술의 접목은 기존의 발전소에도 영향을 주어 K.H. Fasol은 최근의 모델링, 시스템 해석, 모의시험을 통하여 이미 사용 중인 노후 발전소의 리엔지니어링을 통하여 보다 안정적이고 효율적으로 운용이 가능하다는 것을 입증하였다^[6]. 또한, 현장에서의 단순한 설비 운용에서 벗어나, 좀 더 자세한 동역학적인 해석을 하고자하는 노력이 생겨나고, 다시 설비의 개량을 통하여 개선되는 특징을 보이고 있다^{[7]~[9]}.

발전소에서의 조속기의 경우도 전기전자산업의 발전으로 인해 성능 및 신뢰성을 강화시켜나가고 있다. 이러한 진전된 조속기로 제어시스템을 설계하여 여러 종류의 터빈에 적용할 수가 있을 것이다. 선진국의 경우 적극적으로 디지털 조속기 시스템을 개발하여 성능 및 신뢰성을 강화시켜나가고 있는 추세이다^{[10],[11]}.

1.2 연구의 방법

본 연구에서는 발전기의 터빈을 컨트롤하는 제어시스템을 디지털식으로 개발하기위한 기본적인 설계로 하드웨어적인 면과 소프트웨어적인 면을 설계하고 제어대상의 알고리즘, 모델링, 파라미터를 결정하고, 시뮬레이션을 하여 제어시스템의 유효성을 평가한다.

속도제어에 필요한 시스템의 조속 알고리즘은 Woodward Governor

사의 디지털 조속기용 시스템 알고리즘을 분석하여 수정 보완하고 디지털 조속기가 가져야 할 기능들을 열거하여, 10개의 복합블록과 19개의 단일 블록으로 구성되는 시스템 알고리즘을 개발하였다.

제어대상의 모델링에 대한 연구는 복제주화력발전소의 발전용 증기터빈 시스템으로부터 제어대상의 모델을 얻었다.

운전데이터와 알고리즘을 이용하여 모델의 파라미터를 온라인 추정하는 방법을 제안하고자 한다.

1.3 연구의 구성

본 논문은 개별메모리를 갖고 있는 CPU의 3중화와 I/O의 2중화로 되어 있는 디지털 조속기에 대한 하드웨어와 소프트웨어를 설계하여 안정된 출력 신호로 서보 밸브를 구동하는데 주목적이 있으며, 모델링을 통해 각 계통에서의 전달함수를 구하고 제어대상에 대한 통합 전달함수를 구할 수 있다. 설계된 제어계의 유효성은 발전소에서 실제 조업으로 확인하기에는 설비의 안전상 많은 어려움이 따름으로 MatLab simulink로 시뮬레이션을 하여 유효성을 검증 하였다.

각장에서의 구체적인 내용은 다음과 같다.

제1장에서는 본 논문의 연구배경과 연구의 방법, 연구의 구성에 대해 설명하였다.

제2장에서는 하드웨어 설계에 앞서 기존 조속기의 원리를 검토하고, 전력계통에서의 조속기의 운전방법의 종류에 대한 설명 및 기존제품의 특징을 살펴보고 하드웨어를 개발하였다.

개발된 시스템의 모듈에 대해서는 CPU 모듈을 비롯한 각 모듈별 특징을 설명하였다.

제3장에서는 소프트웨어의 설계로 디지털 조속기는 운영 프로그램과

Governor control에 필요한 응용 프로그램으로 구분된다.

운영 프로그램은 CPU 기동에 필요한 프로그램으로 전원인가 후 하드웨어 리셋에 의해 CPU 내의 프로그램 카운터 초기화와 모든 슬레이브 디바이스를 초기화 한다.

응용프로그램은 조속기 제어에 필요한 입출력 프로세싱, 시스템 shutdown 프로그램 등을 수행한다. 각 CPU는 응용프로그램에 의해 데이터를 처리하고 처리 결과를 재 교환하여 BAD 데이터를 차단 할 수 있도록 하였다.

제4장에서는 시스템에 대한 알고리즘을 개발하여 디지털 조속기시스템의 전체의 흐름을 가늠할 수 있도록 하였다.

제5장에서는 제어알고리즘으로 발전용 증기 터빈의 동특성은 운전조건이나 출력조건 등의 운용조건에 크게 의존하는 복잡한 열역학적 플랜트이다. 발전 중 부하변동 등으로 인한 운전조건의 변화에 따른 제어성능의 열화 등, 최적인전을 위한 요건들을 충분히 고려하여 만족되는 제어계를 설계하기 위하여 우선 제어대상의 모델을 얻는 것이 필수적이다. 특히 발전시스템은 분담할 부하의 크기에 따라 시스템의 파라미터 변화가 극심하므로 전 운전조건에서 양호한 모델을 얻는다는 것은 상당한 어려움이 있다. 따라서 본 연구에서는 빈번하게 운전되는 부하점에서 제어대상의 모델 구조를 얻어 선형화하고 파라미터 값은 실제 터빈의 운전 데이터를 수집하여 각각의 동작속도에서 주어진 응답특성을 만족하는 PID제어를 행하고 조속되도록 한다.

제어대상의 모델링과 파라미터 결정은 각 계통에서 개별 전달 함수를 구하여 통합 전달 함수를 결정한 후, HGA를 통해 관측된 데이터를 이용하여 목적 함수 값을 최적으로 하는 파라미터들을 추정한다.

제어기로는 PID 이론을 적용하여 한계감도법과과 릴레이 피드백에 의한 이득 결정방법으로 하여 K_P , T_I , T_D 를 구하고 이상의 결과를 이용하여 설계되어진 제어기의 유효성을 시뮬레이션을 통해 확인하였다.

제6장에서는 본 연구의 결론에 대해 언급하였다.

제 2 장 하드웨어의 설계

2.1 조속기 원리

조속기는 최소한 두 가지의 구성요소 즉, 속도감응요소(Speed sensing element)와 원동기의 입력제어장치(Throttle valve, Steam control valve 등)를 조작시키는 장치를 포함한다. 원동기의 입력을 조정하는 장치를 동작시키는데 비교적 큰 동력이 소요될 경우에는 서보모터(Servo-motor)라는 장치가 필요하게 된다. 서보모터는 원동기의 입력을 조절하는데 소요되는 동력을 발생시키는 장치이며 속도감응장치에 의해 작동된다. 이처럼 서보모터를 쓰는 이유는 속도감응장치의 출력이 매우 미약하기 때문이다.

2.1.1 속도감응장치

가. 기계식 조속기

그림 2.1은 원심력을 이용하는 속도감응장치로서 일반적으로 응용되고 있는 기계식 조속기이다

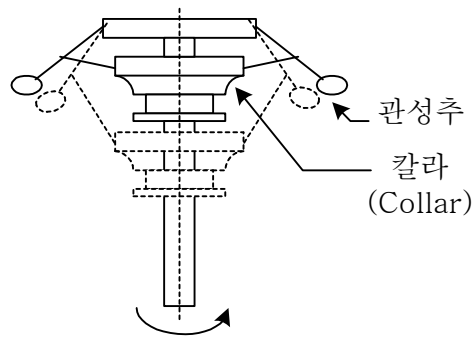


그림 2.1 기계식 조속기

Fig. 2.1 Mechanical governor

이 장치의 원리는 물체가 일정속도의 원운동을 하기 위해서는 힘이 필요하다. 즉, 이 힘은 회전속도의 제곱에 비례하고 회전축에서 원운동을 하는 물체까지의 거리에 비례한다.

회전속도가 상승하게 되면 관성추가 벌어져 가고, 반대로 회전속도가 감소하면 칼라가 아래로 내려오게 된다. 관성추와 칼라는 각 회전속도에 대해 상응하는 독특한 위치를 갖고 있다.

속도 변화에 따라서 칼라의 위치가 상응하여 변하게 되고, 회전속도가 일정하면 칼라의 위치도 일정한 위치에 머물러 있게 된다. 회전속도 변화에 따른 칼라의 상하동작은 기계적으로 범위를 벗어나 관성추가 밖으로 벌어지면 연료 조정 변을 단도록 되어있다.

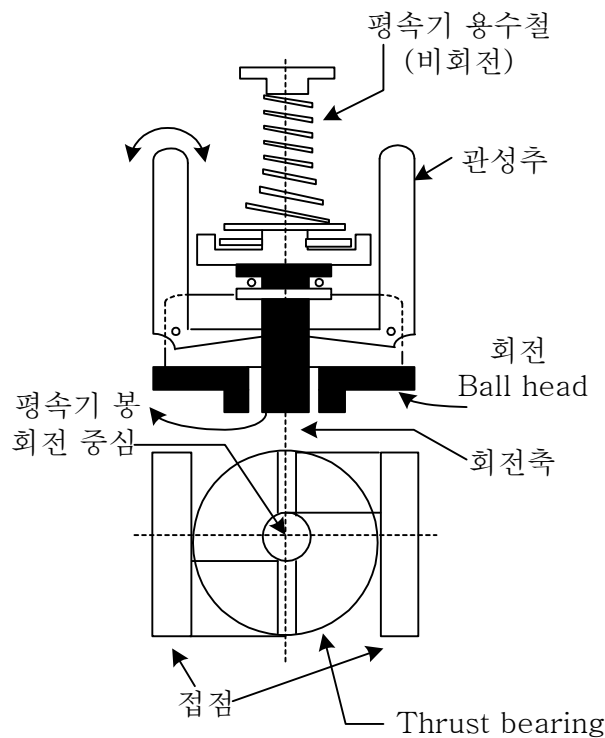


그림 2.2 볼 헤드 속도감응장치

Fig. 2.2 Equipment of speed sensing ball head

그림 2.2는 원심력을 이용한 속도 감응장치의 형태로서, 디젤엔진 발전기, 소용량 증기터빈발전기, 펌프구동용 증기터빈의 속도제어에 사용되는 일반적인 것이다. 이런 속도감응장치를 “Ball head”라고 부른다.

여기서 “Ball head”는 원동기의 회전속도에 비례하여 회전하게끔 기계적으로 연결되어 있다. 그러므로, 원동기 속도의 증감에 따라 관성추의 위치가 밖으로 또는 안으로 움직이게 되고, 이 움직임은 관성추 몸체와 직각을 이루고 있는 받침 발(Toes)에 의해 Thrust bearing을 통해 조종 변 또는 평속기(Speeder)축의 위치 변화를 나타낸다. 관성추의 회전절점에는 마찰을 줄이기 위해 구름 베어링을 사용하였다. 관성추의 회전운동에 의한 원심력은 중력대신에 평속기 용수철의 힘에 의해 평형을 이룬다.

나. 유압식 조속기

유압식 조속기의 속도감지방방법도 레버 및 링크의 연결이 별로 없으므로 그 나름대로 장점이 있고, 온도가 극히 낮을 때는 기름의 점도 변화에 영향을 받는 단점도 있다. 속도감응장치는 원심펌프의 토출압력이 속도의 표시로 이용되는 원심펌프와 벨로우즈(Bellows)로 구성되어 있다. 여기에서 원심펌프의 회전속도는 원동기의 회전속도와 비례하고 펌프의 토출압력 또한 펌프의 회전속도 즉, 원동기의 회전속도와 비례하며, 이 압력의 변화는 벨로우즈 상단의 위치 변화를 일으킨다. 이 변위를 이용하여 원동기의 연료 공급 장치인 연료 가감변을 조작하도록 기계적으로 연결되어 있다. 즉, 원동기(터빈)의 회전속도가 증가하면 벨로우즈내의 압력이 증가되고 이 압력 증가는 벨로우즈 상단이 위로 올라가게 하는 기계적 변위를 발생시킨다. 이 기계적 변위는 원동기에 공급되는 연료의 양을 감소시키는 방향으로 연료 조정변을 작동시킨다.

원동기의 속도증감은 벨로우즈의 압력증감과 비례하므로 벨로우즈 상단의 위치변화는 원동기의 회전속도변화와 비례하여 대응한다. 따라서 벨로우즈 상단의 변위로 엔진에 공급되는 연료가감변의 개도를 조정한다. 여기서 벨로우즈를 누르는 “Governor spring”의 눌림을 조정하면 원동기의 제어속도나 출력을 조정할

수 있게 될 것이다. 오리피스가 가변형이면, 그 크기를 조정함으로써 벨로우즈의 압력을 변화시켜서 원동기의 제어속도나 출력을 조정할 수 있게 될 것이다.

다. 전기식 조속기

원동기의 회전속도에 비례하여 전압을 발생시키는 영구자석계자와 정류자로 구성된 소형영구자석발전기가 속도감응요소로 응용되었다. 직류발전기 출력전압의 고저에 따라서 솔레노이드의 위치가 위로 올라가거나 아래로 내려가게 된다. 솔레노이드에 연결된 조정변(Pilot valve) 플런저(Plunger)는 원동기 속도에 따라 위치가 변하게 되며, 원동기가 정격속도로 회전되고 있을 때 조정변 플런저의 위치는 압유의 유로를 차단하는 중립위치에 있도록 조종되어 있다. 영구자석발전기 대신에 마그네틱픽업을 이용한 것이 최근의 전자식조속기의 속도감응부이다.

전자식 조속기의 경우 가장 기본적인 속도의 측정 방법은 센서나 발전기에서 나오는 주파수를 측정하는 것이다. 이러한 신호는 주파수를 가진 교류 신호이고 원동기의 속도와 직접적인 관련은 적다. 측정된 신호는 원동기의 속도에 비례하는 직류전압으로 변환되게 된다. 즉, 빠른 회전을 하는 원동기일수록 큰 직류전압을 가지게 되고 이 전압이 원동기의 실제 속도를 나타내게 된다. 이는 기계 유압식 조속기에서 플라이 웨이트의 원심력과 같은 역할을 하게 된다. 속도의 설정값은 직류전압 레귤레이터가 연결된 전원에 연결되고 이는 기계 유압식 조속기에서의 평속기(Speeder) 스프링과 비교될 수 있다^[12]. 일반적인 전기전자식 조속기에서는 주파수/전압 컨버터가 주류를 이루고 있다^{[13],[14]}.

2.1.2 원동기 입력제어장치

수차의 수량 조정익의 경우 속도 감응장치의 출력으로 수량 조정익을 직접 조작한다는 것은 비실용적이라는 것이 명백하다. 초창기 수차용 조속기에서는 볼헤드에 의해 선택적으로 동작되는 클러치를 통해서 제어되

는 수차로부터 필요한 힘을 기계적으로 취했다. 필요한 제어력이 점차 커지자 좀 더 다른 제어에 대한 요구에 따라 빠른 속도로 수량 조정익을 동작시킬 필요가 발생하여, 이런 간단한 클러치 장치로는 한계에 부딪혔다. 심지어 그 범위 내에서 필요한 제어력도, 이 장치는 부동대를 작게 유지시키기 어렵고 가장 중요한 제어속도가 원동기의 속도편차의 크기와는 관계없이 원동기의 회전속도에 비례하는 등 제어용 서보로서 불리함이 있다. 특히, 제어속도가 원동기의 속도와 비례하는 특징은 작은 속도편차를 너무 지나치게 수정하려고 하는 경향이 있기 때문에 안정성 있는 운전을 어렵게 하였다. 서보의 동작속도는 적어도 작은 속도편차에 대해서는 속도편차의 크기에 비례하는 것이 바람직하다. 이에 대한 대략적인 해소책은 수차가 제속도에 접근하도록 클러치가 접속되는 시간율을 감소시켜 간헐적으로 동작시키는 것이었다. 그러나 이것은 이 장치의 복잡함을 가중시켰고 클러치의 마모를 심하게 하였다.

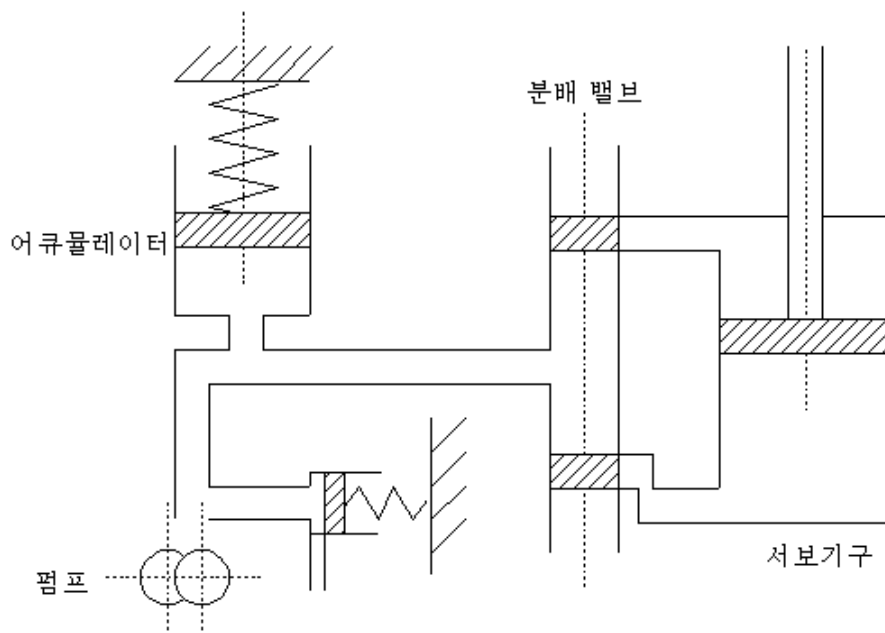


그림 2.3 어큐물레이터와 서보기구

Fig. 2.3 Accumulator type oil pressure pump and servomechanism

어큐물레이터없이 유압서보와 원동기에 의해 돌아가는 유압펌프의 조합은 비교적 값도 싸고 중량도 가벼우며 제어에 필요한 동력을 제공 할 수 있다. 그리고 그림 2.3에서와 같이 어큐물레이터를 갖추면 압유펌프의 용량을 증대시키지 않고도 짧은 시간동안에 서보모터 전행정을 1회 또는 2회 동작시킬 수 있는 아주 높은 수준의 동력을 공급할 수 있게 된다.

어큐물레이터를 갖는 것은 펌프가 전동기에 의해 구동될 때나 제한된 양의 기름이 과열되는 문제가 있을 때는 매우 중요한 것이다. 유압서보의 제어는 피스톤의 밸브를 써서 별로 힘을 들이지 않고 쉽게 할 수 있다. 이 밸브 플런저와 슬리브 사이에 상대적인 회전운동이 도입되면 밸브가 움직일려고 할 때 저항하는 마찰력을 충분히 줄일 수 있기 때문에 출력이 약한 속도감응장치로도 제법 크기가 큰 서보를 빠른 속도로 작동시킬 수 있다. 또한 움직이는 부분의 질량이 이용할 수 있는 힘에 비해 작기 때문에 유압서보의 반응이 극히 빠르다. 작동유체로서 기름을 사용하면 윤활이 자연스럽게 해결된다.

2.2 조속기 운전 방법 및 특징

전력생산기계들의 병렬운전에서 고려되어야 하는 기본사양의 하나는 부하의 적절한 배분이다. 이것은 조속기계통의 제어특성으로 나타나는 스피드드롭을 조속기에 부여함으로써 달성할 수 있다.

교류발전기를 통해 병렬 연결되어 있을 때는 같은 속도로 또는 교류기들의 극수가 각각 다를 때는 비례하는 속도로 회전될 수밖에 없다. 전력계통에서 조속기 운전방법의 형태는 단독부하(Isolated load) 등시성운전(Isochronous), 스피드드롭운전(Speed droop), 연료 조정변을 고정하는 부하제한운전(Load limit) 등이 있다.

여기서 스피드드롭은 엔진 부하가 서보의 개도나 메인스팀밸브개도 변위와 정비례하는 것으로 간주하고 속도에 대한 부하의 궤적으로 나타낸다.

2.2.1 단독부하 등시성운전

발전기 한 대 용량이하의 부하를 담당하는 운전을 단독부하운전이라고 하며 이 경우 부하의 모든 범위 내에서 항상 정해진 주파수를 유지하려면 발전기의 조속기는 항상 일정한 제어속도로 운전되어야 한다. 이와 같이 부하의 변동에 관계없이 발전기의 속도를 항상 일정하게 유지하는 운전방식을 등시성운전이라고 하며, 조속기의 제어속도가 일정하게 유지되는 조속기를 등시성 조속기라고 한다. 스피드드롭 조속기에서 스피드드롭을 0으로 조정하면 등시성 조속기가 된다. 등시성 조속기의 동작을 수평한 선들의 모임으로 특징을 표시 할 수 있으며 각 선은 조속기 속도설정값 (Speed setting)을 나타낸다.

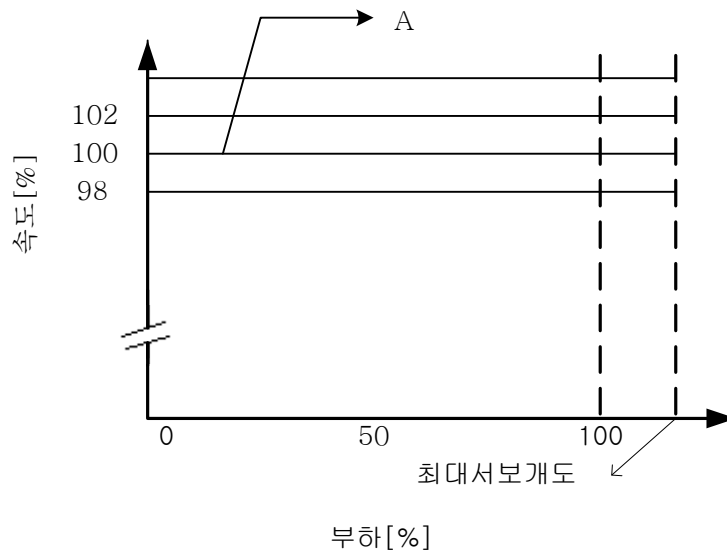


그림 2.4 단독부하 등시성 운전

Fig. 2.4 Isochronous mode at isolated load

그림 2.4에서 선 A로 표시되는 속도를 유지하도록 조속기의 속도를 설정하고, 비동기성(Nonsynchronous) 부하에 연결하여 부하가 증가되면 속도는 무부하로부터 증가됨에 따라 엔진의 최대용량에 도달할 때까지 속도는 선A에 일정하게 머물러 있을 것이고, 부하가 더 증가되면 그때는 메인 스팀밸브개도는 최대에 머물러 있으면서 속도는 떨어질 것이다.

2.2.2 단독부하 스피드드롭 운전

스피드드롭 조속기도 비슷한 선들을 갖고 있으며 그림 2.5에서와 같이 부하변화에 따른 속도를 나타내는 이 선들이 기울어져 있고, 속도조정이 크면 클수록 이 선들의 기울기가 커진다.

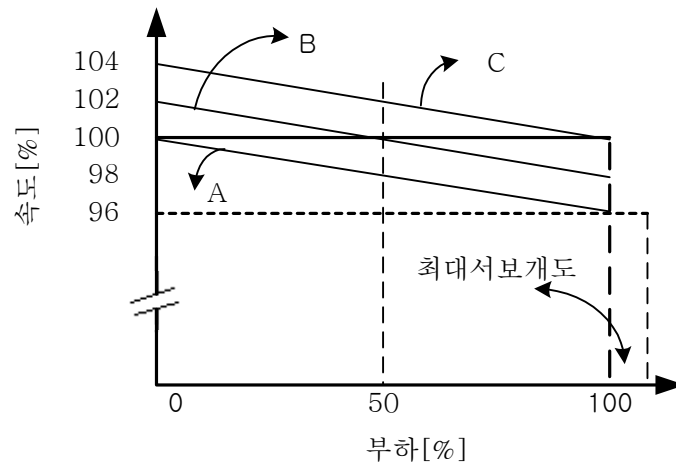


그림 2.5 단독부하 스피드드롭운전

Fig. 2.5 Speed droop mode at isolated load

무부하에서 부하를 증가시키면 연료 조정변 최대개도까지는 레귤레이션 선을 따라 속도가 직선적으로 떨어지게 될 것이다.

여기서 부하제한장치를 50[%]에 맞추어 놓은 경우를 살펴보면 속도가

연료조정변 개도 최대까지 부하가 증가되기 전에 부하제한이 설정되어 있는 50[%]에서부터 속도가 떨어진다.

2.2.3 무한전원 스피드드롭운전

스피드드롭 발전기가 무한전원에 연결되어 있다고 가정해 보자. 이렇게 되면 조속기의 속도에 따라 터빈의 속도가 결정되지 못하고, 동기화 토크가 최대를 넘지 않는 한, 모선의 주파수에 의해 좌우된다. 이경우 조속기의 속도를 변경시키면 발전기의 속도가 변하지 않고 출력(부하)이 변한다. 그림 2.5에서 계통에 연결하기 위해 발전기를 동기시키려면 무부하시 속도가 모선의 속도와 같은 A선을 택해야 한다. 일단 계통에 연결시킨 후 조속기 속도를 A에서 B로 증가시키면 속도가 증가되지 않고 약 50[%]의 부하가 걸리게 된다. C까지 조속기 속도를 점차 증가시키면 발전기에 100[%] 부하가 걸리게 된다. 속도조정이 그림 2.5에서와 같이 4[%]이고, 50[%]의 부하가 걸려있는 상태(조속기가 B에 맞추어져 있는 상태)에서 이 발전기를 모선에서 분리시키면, 아무 구속을 받지 않는 발전기는 안정상태의 속도가 100[%]보다 2[%] 높은 속도로 상승될 것이다. 또한 부하가 걸려 있을 때 즉 조속기 속도가 C에 설정되어 있을 때 발전기를 모선에서 분리시키면, 무부하시 속도는 동기속도인 100[%] 속도보다 4[%]가 높은 104[%]가 될 것이다.

2.2.4 무한전원에서 등시성 운전

만일 스피드드롭이 0이라면 즉 등시성 조속기라면 이런 호기를 무한전원에 연결하는 것은 비실용적이다. 왜냐하면 속도설정을 모선의 속도와 정확하게 일치시키는 것이 실질적으로 불가능하기 때문에 조속기의 속도가 모선의 속도보다 높게 설정되면 비록 차이가 작을지라도 실제 계통의 속도가 조속기의 속도보다 낮으므로 조속기는 엔진이 출력을 계속 줄이는 방향으로 동작하여 연료 조정변을 완전히 닫히게 하고 그로 인해 발전기는 전동기화 되기 때문이다. 그러므로 조속기로 제어되는 원동기의 용량

이 계통의 속도에 영향을 미칠 수 없는 작은 경우에 계통에 병렬 연결시킬 때 조속기의 스피드드롭을 0으로 등시성으로는 맞추지 말아야 한다.

2.2.5 소용량 부하 담당 운전

가. 동일 용량 동일 스피드드롭의 2기 운전

같은 크기의 스피드드롭을 갖고 있는 동일한 1, 2호기가 서로 병렬 연결되어 있고 계통에는 다른 발전설비가 없다면 계통의 주파수는 이들 동일한 두 호기 각각의 조속기 속도 설정값과 조속기-엔진 조합의 속도제어특성 및 계통 전체의 부하에 따라 결정된다.

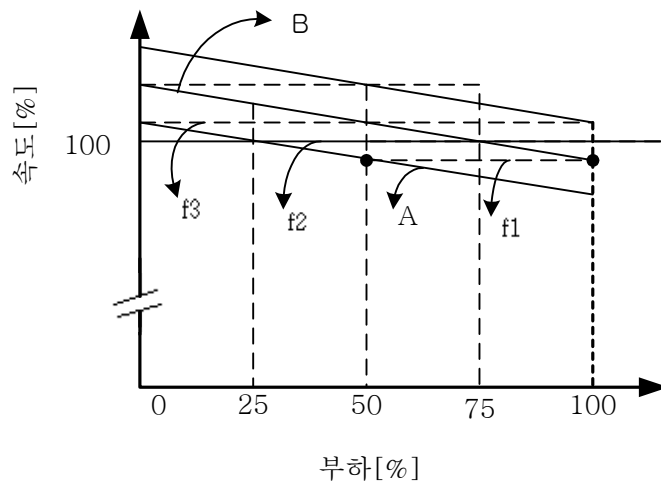


그림 2.6 동일 용량 동일 스피드드롭의 2기 운전

Fig. 2.6 Same capacity and speed regulation (at 2 sets operation)

동일한 1, 2호기의 속도가 그림 2.6에서 A에 맞추어져 있고 계통의 총 부하는 1호기의 용량에 해당하는 100[%]라고 하면 각 엔진은 동일한 1, 2호기가 다 같이 50[%]의 부하를 담당하고 속도는 f1이 된다. 여기서 동일한 1,2호기의 속도조정을 4[%]로 가정하고, 1호기의 조속기 속도는 그대로 A에 두고, 다른 1호기의 조속기 속도를 A보다 2[%] 높은 B로 상승

시키면, 동일한 1, 2 호기가 전체부하 100[%]을 감당하며 운전되는 속도는 f_1 보다 1[%] 높은 f_2 가 되며 부하는 조속기속도가 높은 호기는 75 [%]를 조속기 속도가 낮은 호기는 25[%]를 담당하게 되어 결과적으로 부하가 균등하지 않게 분담된다.

그러나 계통의 속도상승은 한 호기의 조속기속도 증가분인 2[%]만큼 되지는 않는다. 따라서 1, 2호기간의 용량이 같기 때문에 1호기에서 변화시킨 조속기속도 증가분의 반(1/2) 만큼 증가 될 수 있다.

즉, 1호기의 조속기속도는 A 에 있고 다른 1호기의 조속기속도는 B인 상태에서 부하가 150[%], 즉, 50[%]가 증가되면 1, 2호기 각각의 속도조정이 같지 않기 때문에, 증가분 50[%]은 1, 2호기에 균등하게 분담되어 조속기속도가 A인 호기는 50[%]의 부하를 B인 호기는 100[%]의 부하를 담당하게 되며 속도는 다시 f_1 이 된다. 그렇지만 부하가 더 증가되면 조속기 속도가 B인 호기에 과부하가 걸리므로 이 계통의 유효용량은 1, 2호기의 조속기속도 설정값이 동일하다면 1, 2호기 용량의 합인 200[%]이 될 수 있겠지만 그렇지 못하고 150[%]밖에 되지 않는다. 부하가 50[%]으로 낮아지면 이 부하는 모두 조속기속도가 B인 호기에 걸리고 A는 무부하가 되고 주파수는 f_3 가 된다. 속도조정이 동일하지 않아도 부하의 분담이 부적절하게 된다.

나. 스피드드롭이 다른 동일용량 2기 운전

그림 2.7과 같이 속도조정이 다른 1, 2호기 (조속기 속도가 A인 호기는 2[%], B인 호기는 4[%]로 가정)가 각각 50[%] 부하를 담당하도록 조속기 속도가 설정되어 주파수 f_1 으로 운전되고 있는 상태에서 부하가 150[%]으로 증가되면, 증가된 부하 50[%]의 3분의 2는 속도조정이 낮은 조속기 속도가 A인 호기에, 나머지 3분의 1은 속도조정이 4[%]인 조속기 속도가 B인 호기에 더해지고 새로운 주파수 f_2 로 운전될 것이다. 여기서 속도조정이 높은 조속기 속도가 B인 호기의 조속기 속도를 조정함으로써 호기 각각의 부하는 75[%]를 유지하면서 임의의 속도를 유지하

계 할 수 있다.

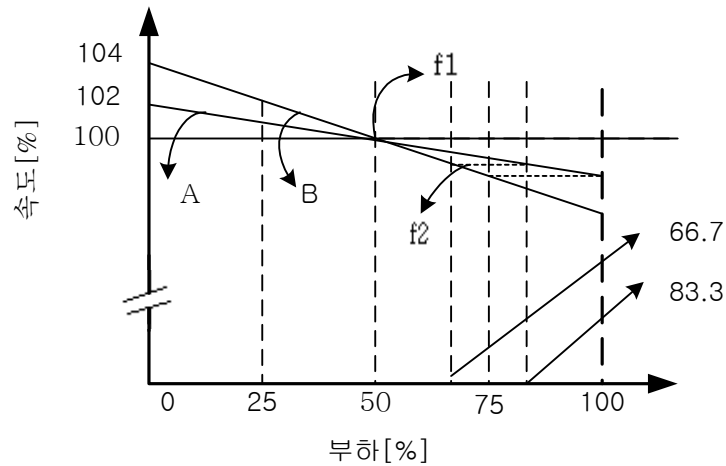


그림 2.7 스피드드롭이 다른 동일 용량2기운전
 Fig. 2.7 Same capacity and different speed regulation
 (at 2 sets operation)

다. 2기중 1기의 스피드드롭이 0일 때

1,2호기의 속도조정이 각각 다른 운전의 경우는 1호기의 속도조정이 0인 경우이다. 이 경우 전원계통의 주파수는 등시성으로 운전되는 호기의 용량범위 안에서는 그 호기의 조속기속도 설정값을 유지하여 무한모션의 경우와 동일하다. 모든 영구적인 부하의 변화는 등시성 호기에 걸리게 된다. 각각 50[%]의 부하를 담당하고 있는데 부하가 150[%]으로 증가되면 스피드드롭을 갖고 있는 호기의 부하는 50[%]에 머물러 있고 부하의 증가분은 모두 등시성 호기에 걸려 이 호기의 출력은 100[%]이 된다. 부하가 더 늘어나면 등시성 호기는 과부하가 되고 계통의 주파수가 떨어지는 만큼 스피드드롭을 갖고 있는 호기가 담당하게 된다. 또, 전체 부하가 50[%]으로 감소될 경우에는 등시성 호기는 무부하가 되고 50[%]의 부하는 모두 스피드드롭을 갖고 있는 호기에 걸리게 될 것이다. 부하가

50[%]보다도 더 감소되면 계통의 주파수는 상승되고, 등시성 호기는 스피드드롭을 갖고 있는 호기에 의해 그 조속기속도 설정값보다 높은 속도로 전동기화 될 것이다. 실용적으로 이런 운전은 계통의 주파수를 일정하게 유지시킬 수 있는 이점이 있고, 극단의 경우는 등시성 호기의 출력이 가능하면 50[%]에 유지되도록 스피드드롭을 갖고 있는 호기가 부하의 변동분을 감당하도록 한다.

위에 설명된 사항은 속도만 감지하는 조속기에서의 경우다. 조속기에 부하감응장치가 있는 경우에는 1,2호기를 병렬 연결하여 등시성운전을 하면서도 적절한 부하를 분담하는 운전이 가능하다. 그런 조속기를 부하분담조속기(Load sharing governor)라고 한다.

2.3 하드웨어 설계

2.3.1 개요

하드웨어는 CPU, SIO, Digital speed sensor, Analog input, Analog output 및 Discrete input 모듈로 구성되어 있다. 개발 완료된 것은 입·출력 제어 및 연산처리를 하는 CPU 모듈과, MPU로부터 입력되는 데이터를 받아 속도를 검출하는 Digital speed sensor 모듈, Actuator를 구동하는 아날로그 출력모듈 그리고 키패드, LCD, RS232회로로 구성된 SIO 모듈이다.

각 모듈의 설계기준은 안전성, 신뢰성, 확장성을 고려하여 개발하였다. CPU 모듈은 2 out of 3의 voting 방식을 구현하여 신뢰성을 높였다. 또한 지금까지 연구 개발된 것으로 하나의 loop를 구성하여 모터를 이용한 LCD 혹은 컴퓨터로 데이터를 디스플레이하여 분석할 수 있다.

2.3.2 시스템 계통

디지털 조속기 시스템의 전체 계통도는 그림 2.8과 같이 CPU의 3중화와 I/O의 2중화로 되어있다. 3개의 CPU 모듈은 개별 메모리를 갖고 있으며 외부 센서로부터 입력된 신호를 각각 연산처리 후 데이터를 2 out of 3 voting을 사용하여 안정된 출력 신호로 서보 밸브를 구동한다.

CPU A와 CPU C는 입력 디바이스로부터 독립적으로 값을 받으며, 터빈의 속도와 같이 중요한 입력은 두 개의 독립적인 센서로부터 값을 받는다. 각 CPU로 입력되는 데이터 값은 직렬 통신 제어기에 의해 공유하게 되며, 각 CPU가 voting을 수행하도록 하였으며, 그림 2.9와 같이 구성되어 있다.

MPU는 회전수에 비례하는 펄스신호를 발생시키며, 이 펄스 신호를 I/O 모듈에서 정형화 하여 표준 디지털 신호로 변환시킨 후 CPU가 회전속도를 연산처리 하게 된다.

디지털 입력 신호는 매트릭스형의 회로에 신호를 인가한 후 이를 스캐닝하여 인터록이나 각종 리미트 스위치의 신호를 CPU가 연산처리 후 출력한다.

LVDT는 터빈의 밸브개도를 검출하는 장치로서 LVDT형이 온도에 대한 내력이 우수하며 밸브를 구동하는 서보 실린더의 스피들이 LVDT를 연동으로 구동하도록 구성하여 밸브개도의 아날로그 신호를 디지털 신호로 변환시킨 후에 이 신호가 CPU로 전달된다.

Key pad는 조작자가 필요시 프로그램의 변경이나 기타 조작을 할 수 있는 입력장치이며, 매트릭스 구조로 CPU가 스캐닝을 한 후 입력된 key를 reading 하도록 하였다.

디지털 출력은 Relay board를 통하여 a 점점 또는 b 점점으로 출력되어서 필요한 디지털 신호를 출력하도록 구성되었고, 아날로그 신호는 아날로그형 계기의 출력으로 이용케 한다.

local에 데이터 전송을 위하여 RS232C를 통하여 디지털 조속기 시스템이 운용될 수 있도록 구성하였다.

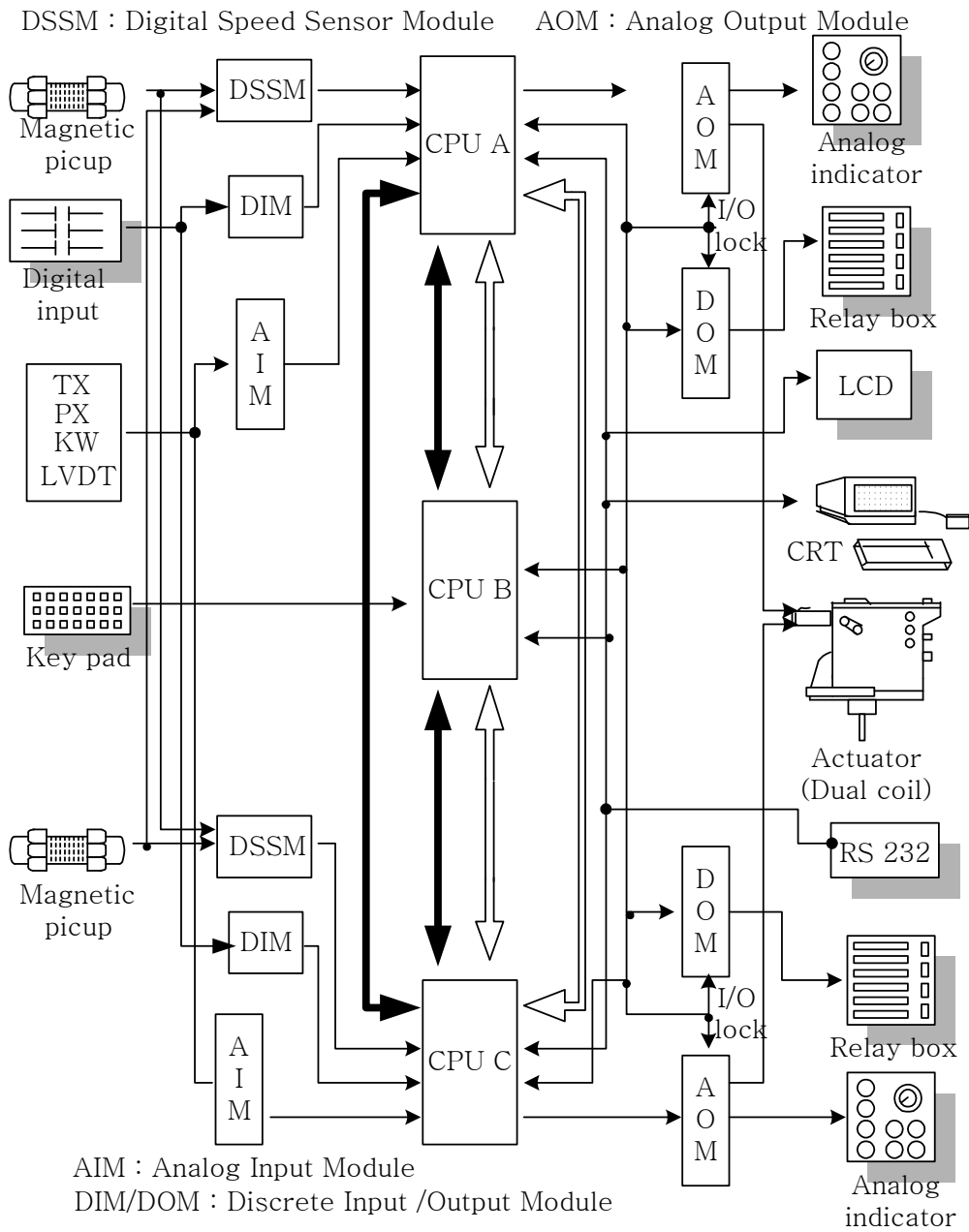


그림 2.8 디지털 거버너 개념도

Fig. 2.8 Configuration of the digital governor system

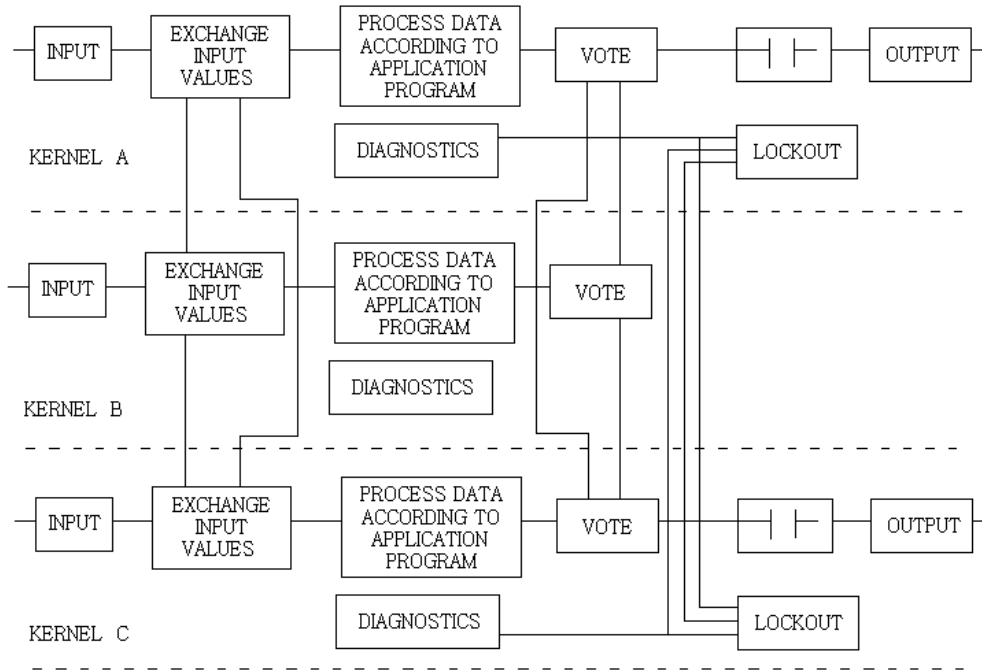


그림 2.9 시스템 계통도

Fig. 2.9 Block diagram of the system

2.3.3 개발 시스템 모듈별 설명

가. CPU 모듈

(1) 개요

개발된 CPU 모듈은 MC68EC040을 사용하여 모듈 자체의 높은 성능을 보장하였으며, Serial communication controller를 통하여 CPU간 데이터 전송을 안정하게 할 수 있게 하였다. 주 메모리는 데이터용과 명령용으로 분리하여 CPU의 명령 실행속도를 향상시켰다.

CPU는 시스템 초기화시 Mother board의 CPU 모드를 결정하도록 하여 (A CPU=00 B CPU=01 C CPU=10) CPU간 데이터 교환 및 I/O 제어에 사용하도

록 하였다.

(2) 특징

- ① MC68EC040 32bit 마이크로프로세서
- ② Data SRAM과 Instruction SRAM의 분리
- ③ Inter CPU communication 기능
- ④ Battery backup 기능(Back용 SRAM)

(3) CPU 모듈 어드레스 맵

표 2.1과 같이 CPU 모듈의 어드레스 맵을 할당하였으며 I/O는 확장 가능하게 설계하였다.

표 2.1 어드레스 맵

Table 2.1 Address Map

ADDRESS	CAPACITY	유효 데이터용역	액세스 단위	ASSIGNMENT
00000000~000ffffc	256k BYTE	D31~ DO	LONG WORD	EPROM
02000000~021ffffc	512 BYTE	D31~ DO	BYTE,WORD,LONG WORD	데이터용 SRAM
03000000~031ffffc	512 BYTE	D31~ DO	BYTE,WORD,LONG WORD	명령용 SRAM
04000000~0403ffffc	8K BYTE	D31~ DO	LONG WORD	백업용 SRAM
05000000~05ffffff		D31~ DO	LONG WORD	입, 출력
05000000		D7~ DO	LONG WORD	REALIME CLOCK
~05100000		D7~ DO	LONG WORD	타이머
~05200000		D7~ DO	LONG WORD	시리얼 통신
~05300000		D7~ DO	LONG WORD	컨트롤&스테이터스레지스터
~05400000		D7~ DO	LONG WORD	IO #0
~05500000		D31~ DO	LONG WORD	IO #1
~05600000		D31~ DO	LONG WORD	IO #2
~05700000		D31~ DO	LONG WORD	키 패드
06000000~ffffff		D31~ DO	LONG WORD	RESERVED

(4) 회로 설명

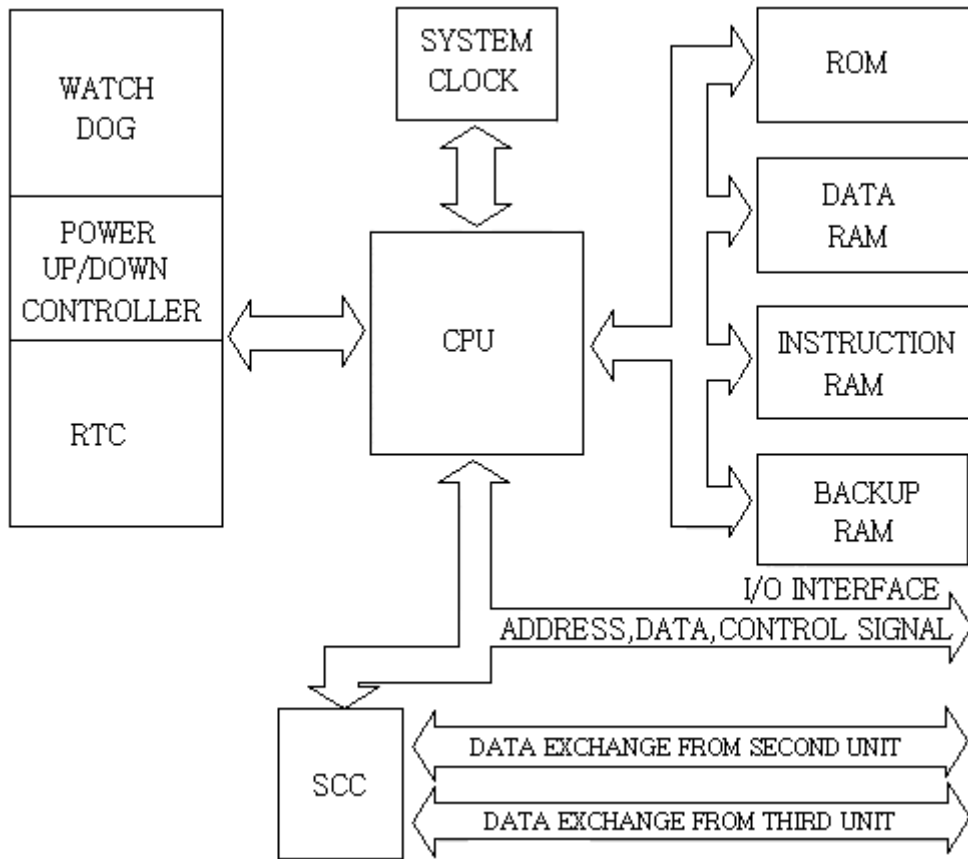


그림 2.10 CPU 모듈 내부 구조

Fig. 2.10 Block diagram of CPU module

① 마이크로프로세부

CPU는 MOTOROLA사의 MC68EC040을 사용하고 있으며 MC68EC040 프로세서는 MOTOROLA 사의 HCMOS 기술로 제작된 고성능, 고속 프로세서로 다음과 같은 특징이 있다.

- MC68EC040과 호환되는 Integer execution unit
- 독립된 4K Byte 명령용 캐시와 데이터용 캐시
- 32 비트 Non - multiplex 외부 어드레스와 데이터 버스 인터페이스
- Bus snooping 을 통한 multimaster / multiprocess 기능
- 4G Byte 의 직접 어드레싱 방법

② 시스템 클록부

설계된 CPU의 클록은 버스 클록 신호와 프로세서 클록신호가 필요하며, 슬레이브 디바이스에 필요한 클록이 요구된다. 프로세서 클록은 안정성을 고려하여 CPU 최대 클록 보다 낮은 50[MHz] 오실레이터를 사용하였고, 버스 클록은 D F/F 을 이용하여 2분주한 25[MHz]를 사용하도록 하였다. 슬레이브 디바이스에 필요한 클록은 2진 카운터를 이용하여 4분주된 12.5[MHz]와 8분주된 6.25[MHz]를 사용하여 CPU 모듈내의 슬레이브 디바이스에 사용하도록 하였다.

③ 어드레스 디코더 회로부

설계된 CPU의 어드레스 버스는 32비트로 이루어져 16M바이트 어드레스 범위를 갖는다. 이 어드레스 영역은 CPU가 제어할 I/O 및 메모리가 서로 충돌이 일어나지 않도록 어드레스를 할당하여, 상위 어드레스 A31 ~ A24, \overline{TS} (전송시작) 그리고 \overline{TA} (전송인지)를 PLD 디바이스가 입력받아 5개의 어드레스 범위로 맵을 할당하였다. GAL22V10은 어드레스 디코더시, 문제되는 전달시간 및 셋업 타임이 수 [ns]이하로 설계된 CPU의 "A", "C" 특성에 적절하며 어드레스 디코더를 프로그램하여 설계할 수 있으므로 디버깅이 용의하고 회로를 간소화 할 수 있다. 74F138은 어드레스 서브 디코더로 슬레이브 디바이스를 8개로 디코더 하여 CPU 모드내의 슬레이브 디바이스 및 외부 I/O에 사용하도록 하였다. CPU 상태나 기타 I/O포트에 사용하기 위해 74F154 디코더를 사용하여 32채널 I/O 포

트를 확보하도록 하였다. 그림 2.11은 어드레스 디코더 타이밍도이다.

어드레스 디코더 PLD 프로그램은 부록의 1을 참조한다.

EPROM 액세스시 PLD 논리식은,

ROM := ((ADDRESS==^h00) & !(TS&ROM) & TA);

로 이루어져있다.

ROM 칩 선택은 \overline{TS} 가 어서트 되고 어드레스가 0X00XXXXXX 일 때 칩 선택이 나타나고 \overline{TA} 가 어서트되면 무효화 된다.

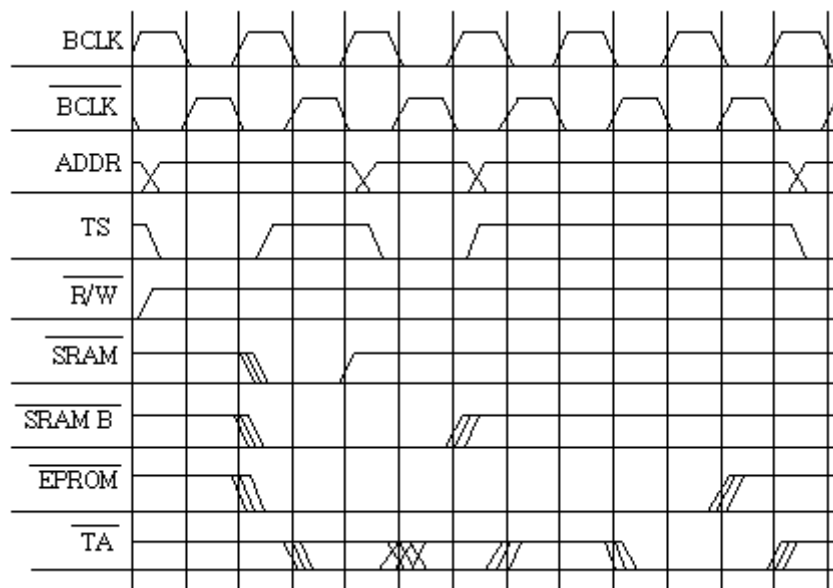


그림 2.11 어드레스 디코더 타이밍도

Fig. 2.11 Time chart of address decoder

④ 웨이트 제너레이터 회로부

동기식 버스는 액세스 타임이 느린 장치와 통신하는 경우 프로세서의 속도를 늦추는 대기 프로토콜을 사용한다. 그림 2.12에서 설계된 CPU의

웨이트 제너레이트는 어드레스 디코더의 모든 출력과 CPU의 제어 신호를 받아 PLD 내에 카운터와 데이터를 설계하여 데이터 통신의 시작과 함께 카운터가 스타트 되어 슬레이브 디바이스에 필요한 웨이트를 삽입한 후 전송인지 신호(\overline{TA})를 어서트 하도록 하였다. 또한, 슬레이브 디바이스의 최대 웨이트 후에도 전송인지 신호가 나타나지 않으면 전송 에러(\overline{TEA})를 나타내도록 설계하였다.

PLD 프로그램은 부록의 프로그램 2를 참조한다.

웨이트 제너레이터 프로그램은 5비트 리플 카운터와 웨이트 수를 설정하여 전송인지를 나타내도록 하였다.

```
BAT := RAMD&RAMI & (COUNT2 #RAMB)
      & (IO #COUNT2 #COUNT3) & (ROM #COUNT0 #COUNT3);
```

이와 같은 PLD 프로그램으로 주 메모리 SRAM을 No 웨이트로 액세스하고 ROM의 경우 액세스타임이 120[ns]이므로, ROM을 안정하게 액세스하기 위해 4웨이트를 삽입하였다. I/O는 현재까지 개발된 슬레이브 디바이스 중 액세스 타임이 가장 느린 리얼타임 클록에 12 웨이트를 삽입하였다.

```
TEA := (COUNT3 # COUNT4) # !TA # !TEA;
```

전송 에러 출력인 TEA는 12 웨이트 후에도 전송인지가 되지 못하면 버스 에러로 판단하여 CPU에게 버스 에러를 출력하도록 하였다. 위 프로그램의 모든 출력은 \overline{TA} , \overline{TEA} 중 하나가 전송인지 후 모든 출력을 “1”인 상태가 되도록 하였다.

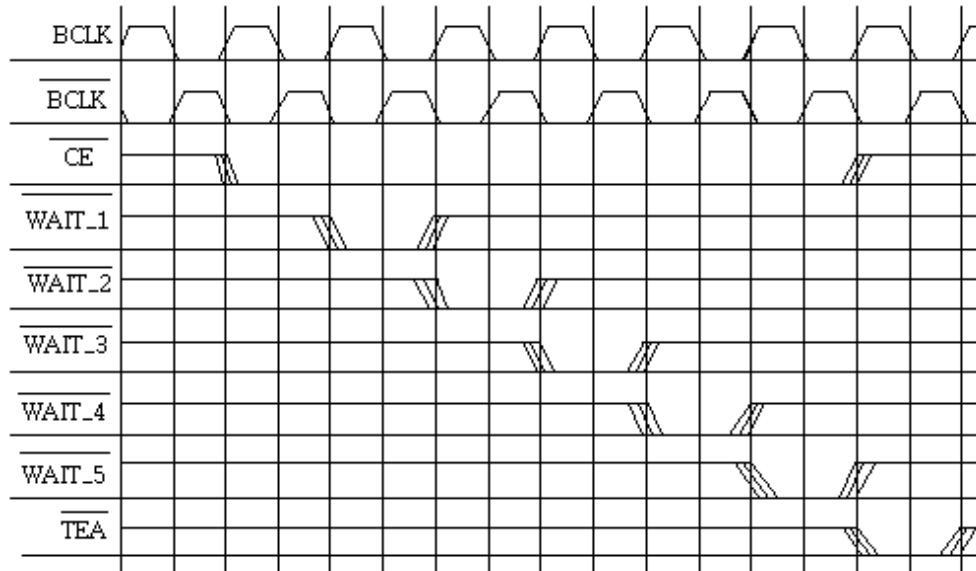


그림 2.12 웨이트 제너레이터 타이밍도
 Fig. 2.12 Time chart of wait generator

⑤ 동적 버스 사이즈 조정 회로부

데이터 버스의 사이즈를 동적으로 조정할 수 있도록 설계된 CPU 모듈은 주메모리인 데이터와 명령용 SRAM에 표 2.2와 같이 버스 사이즈를 동적으로 제어할 수 있도록 하였고 GAL16V8 PLD를 사용하여 로직을 설계하였다.

동적버스 사이즈 제어 PLD 프로그램은 CPU 어드레스 A0, A1과 $\overline{\text{Siz0}}$, $\overline{\text{Siz1}}$ 을 조합하여 표 2.2와 같이 데이터 사이즈를 인코딩 하도록 설계되어 있다.

PLD 로직 프로그램은 부록의 3을 참조한다.

표 2.2 리드/라이트 사이클을 의한 버스 사이즈

Table 2.2 BUS size for READ/WRITE cycle

Transfer Size	Signal Encoding				Active Data Bus Sections			
	Siz0	Siz1	A1	A0	D31~D24	D23~D16	D15~D8	D7~D0
Byte	0	1	0	0	Byte	-	-	-
	0	1	0	1	-	Byte	-	-
	0	1	1	0	-	-	Byte	-
	0	1	1	1	-	-	-	Byte
Word	1	0	0	0	Byte	Byte	-	-
	1	0	1	0	-	-	Byte	Byte
Long Word	0	0	X	X	Byte	Byte	Byte	Byte

⑥ 클록 동기 회로부

설계된 CPU는 2 out of 3 voting을 수행하기 위해 CPU가 3중화로 설계되어 있으므로 3개의 CPU가 한 시스템처럼 동작하기 위해 클록이 동기 되어있다. rate 타이머가 6.25[Mhz]를 입력받아 100[Mhz]마다 펄스를 출력하도록 설계 되어 있으므로, 절연소자를 통해 타이머로부터 출력된 펄스를 각 CPU가 서로 공유하도록 하였고, 공유한 3클록을 PLD 로직을 사용하여 소프트웨어 적으로 가장 적합한 클록을 출력하도록 하였다.

클록 동기 회로의 PLD 프로그램은 부록의 4를 참조한다.

⑦ 주 메모리 회로부

㉠ 주 메모리

SRAM은 CPU 모듈의 주 메모리로서 데이터용과 명령용으로 분리되어 메모리를 효율적으로 관리한다. WMS512K×8의 액세스 시간이 20[ns]로 CPU 와의 데이터 전송시 웨이트 시간 없이 전송 가능하므로 명령 실행 속도를 향상시켰다.

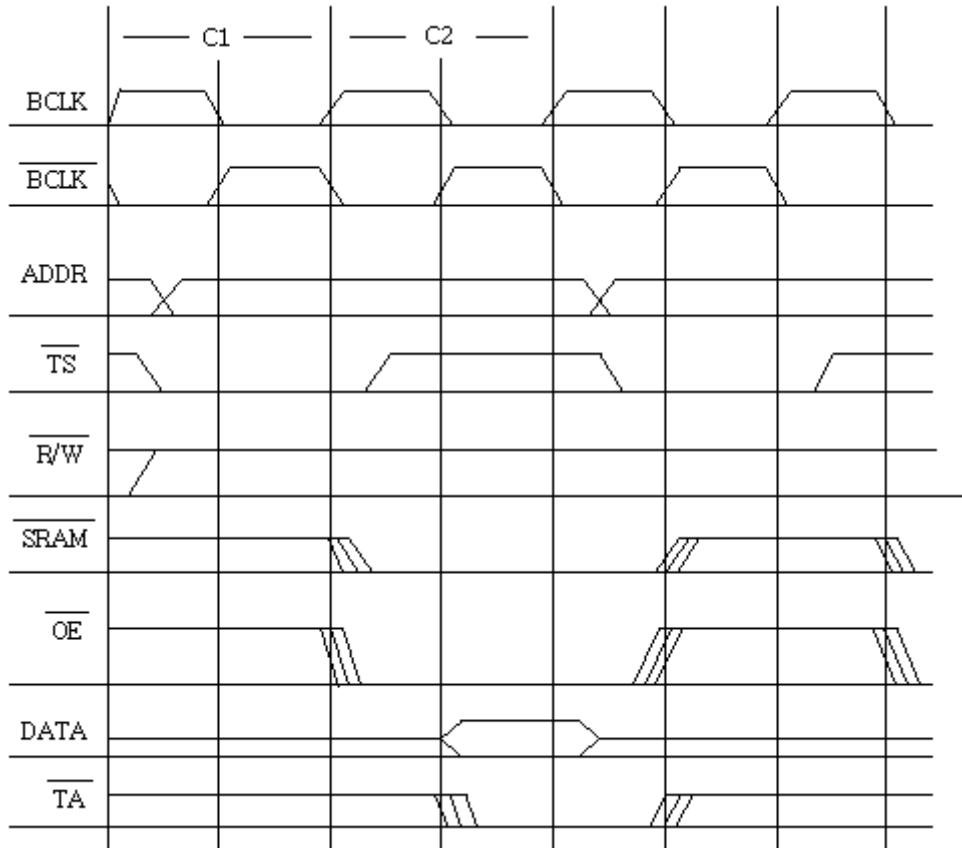


그림 2.13 SRAM 타이밍도
 Fig. 2.13 Time chart of SRAM

㉠ Low power SRAM

Low power SRAM은 CPU 모듈의 데이터 백업 메모리로서 액세스 시간이 45[ns]로 선정하여 데이터 전송시 2웨이트가 필요하며, 파워 다운시 시스템의 중요 데이터와 시스템의 셋 포인터 값을 저장하는 메모리로 사용한다.

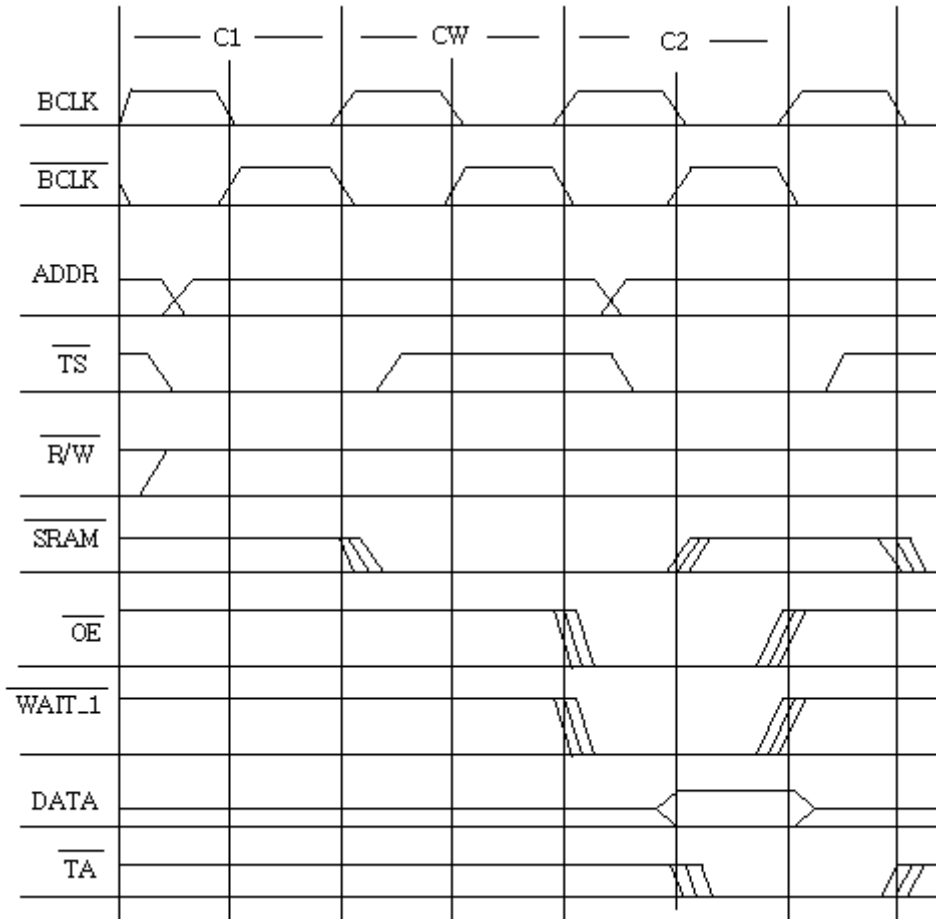


그림 2.14 Low power SRAM 타이밍도
 Fig. 2.14 Time chart of low power SRAM

㉔ 읽기 전용 메모리

EPROM은 CPU 모듈의 OS와 어플리케이션 프로그램을 저장하고 있는 읽기 전용메모리로서 액세스 시간을 120[ns]로 선정하여 데이터 읽기 사이클시 4 웨이트가 필요하다. 파워 업시 EPROM의 데이터로 시스템이 구동되며 구동 후 모든 데이터를 SRAM으로 덤프시켜 러닝 모드시 명령 실행 속도를 최소화 시켰다.

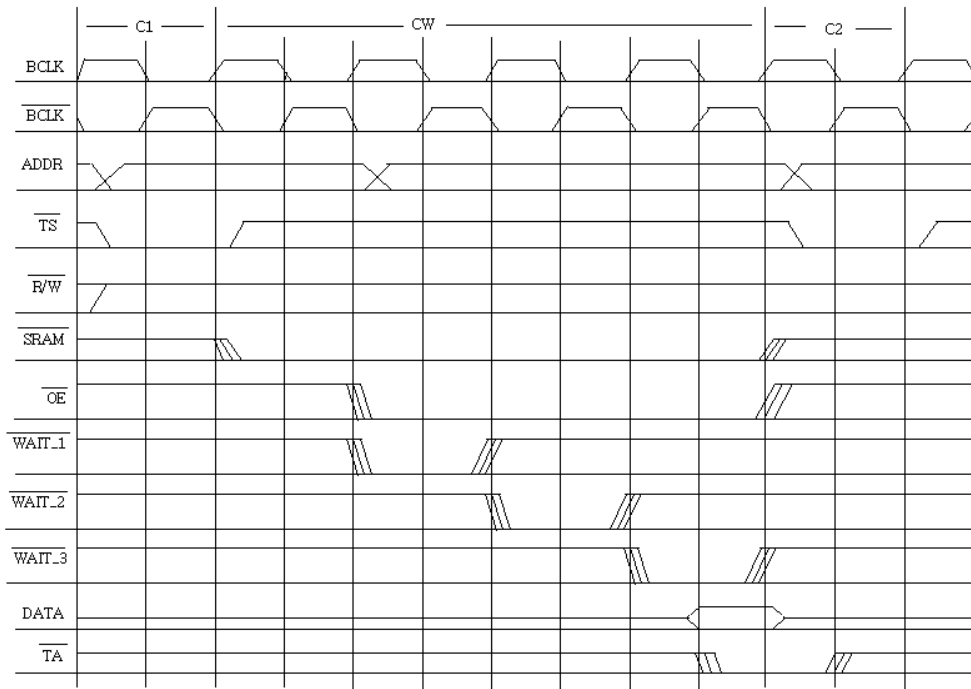


그림 2.15 EPROM 타이밍도

Fig. 2.15 Time chart of EPROM

⑧ 인터럽트 회로부

시스템에서 발생하는 모든 인터럽트는 외적 요인으로 발생되며, 오토백터 인터럽트로 설계되어 있다. CPU는 버스 사이클의 라이징 에지에서 인터럽트 우선순위를 샘플하며 인터럽트 발생시 CPU의 버스 사이클에 동기되어 인터럽트가 인정되면 해당 인터럽트 서비스 루틴을 실행하도록 설계되어 있다.

표 2.3 인터럽트 ASSIGNMENT

Table 2.3 INTERRUPT ASSIGNMENT

우선순위	IPL2-IPL0	ASSIGNMENT	비고
7	000	Power Down	Non-masking
6	001	Speed Sensing	
5	010	Time Schedule	
4	011	Serial Communication	
3	100	Reserved	
2	101	Reserved	
1	110	Reserved	

⑨ 리얼 타임 클록 회로부

설계된 CPU 모듈은 실시간 클록을 이용할 수 있도록 RTC를 사용하였다.

⑩ 시리얼 커뮤니케이션 컨트롤러 회로부

설계된 CPU는 CPU간 데이터 교환을 위하여 Z8530 Serial communication controller를 사용하였다. 속도에 따라 Z8530 CMOS SCC나 Z8530 ESCC로의 업그레이드가 간단하고, 전송 에러 체크가 간단하며, 다중통신이 가능하다. 두 개의 통신 포트 사용이 가능하므로 CPU간 데이터 교환을 하도록 하였고 두 포트는 12.5[Mb]를 입력받아 소프트웨어적으로 보레이트를 결정하여 RS422 동기식 포트 사용하였다. MC34050은 이중 RS422 트랜시버로 SCC의 출력을 받아 다른 두 CPU에 데이터를 트랜시버 하도록 하였다.

⑪ 리셋 회로부

리셋 회로부는 마이크로 프로세서 슈퍼바이저리 회로(MAX696)를 사용하여 리셋 기능 외 배터리 백업 기능, 전원 이상 감지 기능, 워치독 타이머 기능을 갖고 있다.

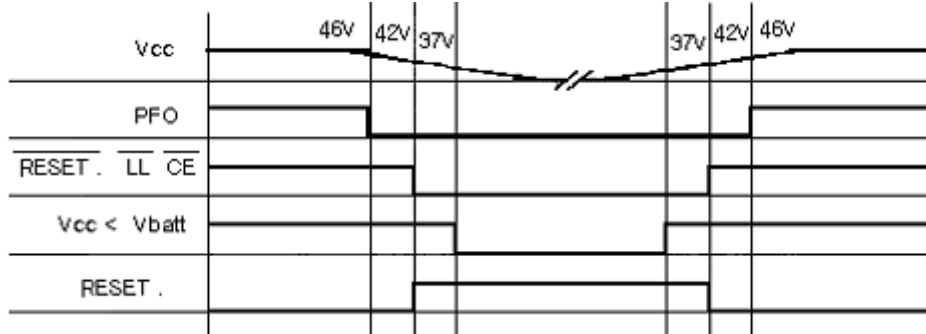


그림 2.16 리셋 타이밍도

Fig. 2.16 Time chart of RESET

㉠ 리셋 기능

MC68040의 경우 전원 인가시 Vcc가 동작 범위 내 이내면서 최소 10개의 $\overline{\text{BCLK}}$ 사이클 동안 어서트된 후 니게이트 되면 CPU는 리셋에 관한 예외처리를 수행하게 된다. 본 모듈의 경우 $\overline{\text{BCLK}}$ 가 40[ns]로 설계되어 있으므로 400[ns]이상이면 리셋 예외 처리를 하며, MAX696의 경우 최소 50[ms] 동안 리셋동작을 하므로 위 전술한 규약에 만족한다.

㉡ 배터리 백업 기능

CPU 모듈의 전원 전압이 리튬 배터리 전압(3.7[V])이하이면 low 파워 SRAM은 리튬 배터리 전원이 인가되어 스탠바이 상태에 있게 된다.

㉢ 전원 이상 감지 기능

CPU 모듈의 전원 전압이 기준 전압(4.2[V])이하이면, 논 마스킹 인터럽트인 파워다운 프로그램을 실행하고, 전원 이상 전압(3.7[V]로 규정) 이하이면 전원 이상 감지를 하여 외부에 지시하도록 설계하였다.

㉣ 위치 독 타이머

입·출력 데이터 전송이 기준 시간(입·출력 최대 전송 시간) 이상이

면 위치독 타이머가 작동하여 그 결과를 외부에 지시하도록 설계하였다.

나. SIO 모듈

(1) 개요

SIO 모듈은 controller에 입력값을 넣기 위한 키패드 회로부와 각종 데이터값을 지시하기 위한 LCD 회로부 및 CRT/MODEM을 위한 직렬통신 제어회로부로 구성된다.

(2) 회로 설명

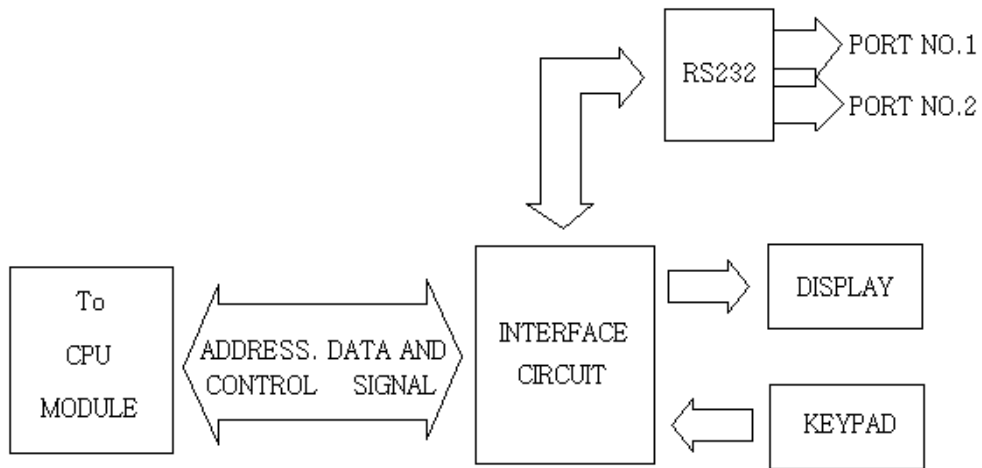


그림 2.17 SIO 모듈 블록 다이어그램

Fig. 2.17 Block diagram of SIO module

① 디코더 회로부

SIO 모듈은 RS232, 키패드 및 LCD 회로로 구성되어 있으며, 할당된 어드레스 맵과 PLD 프로그램은 부록의 5를 참조한다.

표 2.4 SIO 모듈 어드레스 맵

Table. 2.4 SIO module address map

Address	유효 데이터 영역	액세스 단	Assignment
Base + 0000	D24 ~ D0	Long Word	LCD
Base + 1000	D24 ~ D0	Long Word	SCC
*Base = 05400000			

② 키패드 회로부

키패드 회로는 24key(4×6)로 이루어져 있으며, 키패드 패널의 구성은 그림 2.18과 같다

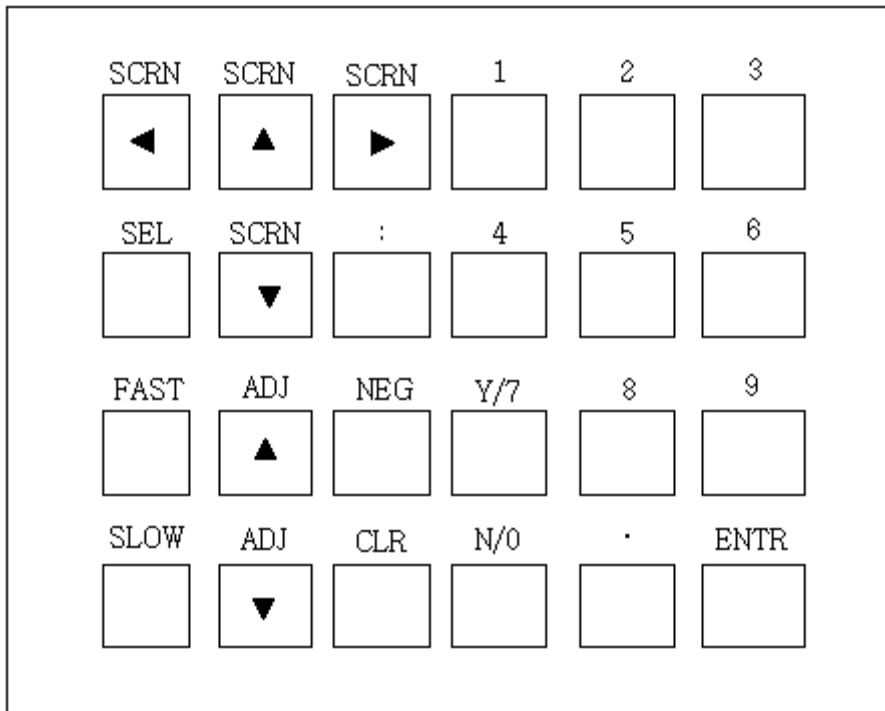


그림 2.18 키패드 구성

Fig. 2.18 KEYPAD

키패드 회로는 키 값이 동시에 "A", "C" 커널로 입력될 수 있도록 하였으며, switch와 Transparent latch로 이루어져있다. 키 입력 값은 10[ms] 주기로 CPU에 의해 polling되어 그 값에 해당하는 동작을 수행토록 설계하였다.

③ LCD 회로

LCD 회로는 그림 2.19와 같이 HD44780 LCD controller를 사용하였고, Character type(40×2)로 설계하였다.

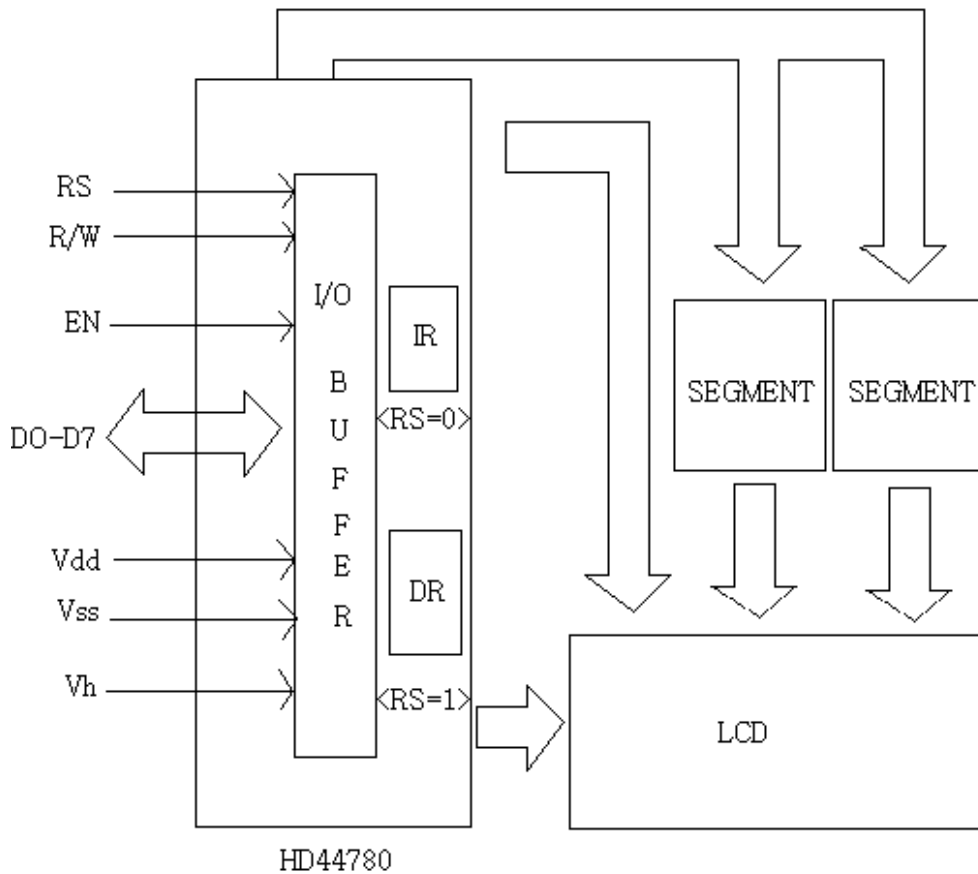


그림 2.19 LCD회로 블록 다이어그램

Fig. 2.19 Block diagram of LCD

LCD controller는 14PIN으로 구성되어 있으며, 각 기능을 정리하면 표 2.5와 같다.

표 2.5 LCD 컨트롤러핀 기능

Table 2.5 Function of LCD CONTROLLER PIN

핀 번호	기 호	레 벨	기 능
1	Vss		접지 (0V)
2	Vdd		전원 (5V)
3	Vo		LCD 밝기 조절
4	RS	H/L	L : 명령 레지스터가 선택 H : 데이터 레지스터가 선택
5	R/W	H/L	L : WRITE (CPU → LCD) H : READ (CPU ← LCD)
6	EN	H	ENABLE 신호
7-14	D0 - D7	H/L	데이터 신호

㉠ LCD 모듈의 입/출력 타이밍

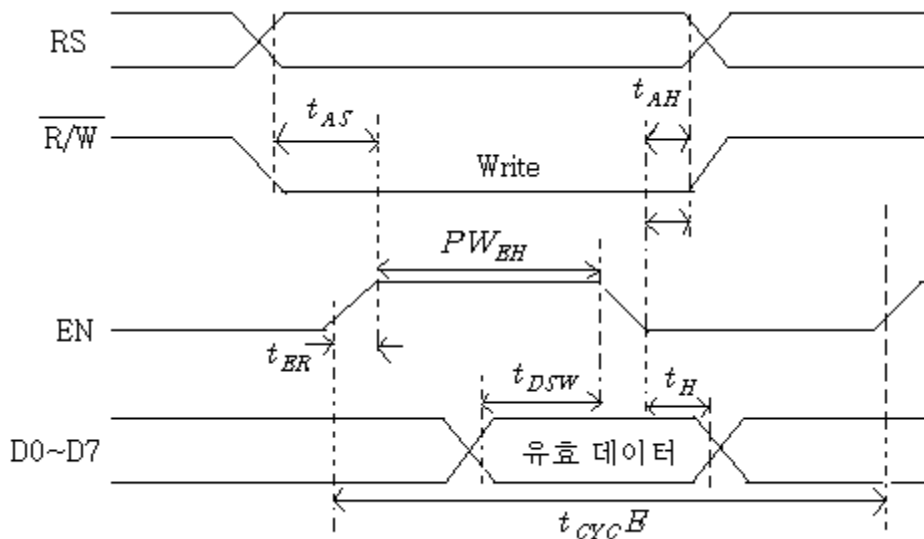


그림 2.20 라이트 타이밍도

Fig. 2.20 Time chart of WRITE

LCD 회로의 라이트 타이밍은 그림 2.20과 같으며, enable 신호가 어서트한 후 니게이트되기전 195[ns]동안 데이터가 유효하며, 니게이트된 후 10[ns]동안 데이터를 홀드 해야만 한다.

LCD 회로의 리드 타이밍은 그림 2.21과 같으며, enable 신호가 어서트 되고 최대 320[ns]이후 데이터가 유효하며, 니게이트된 후 20[ns]동안 데이터를 홀드 해준다.

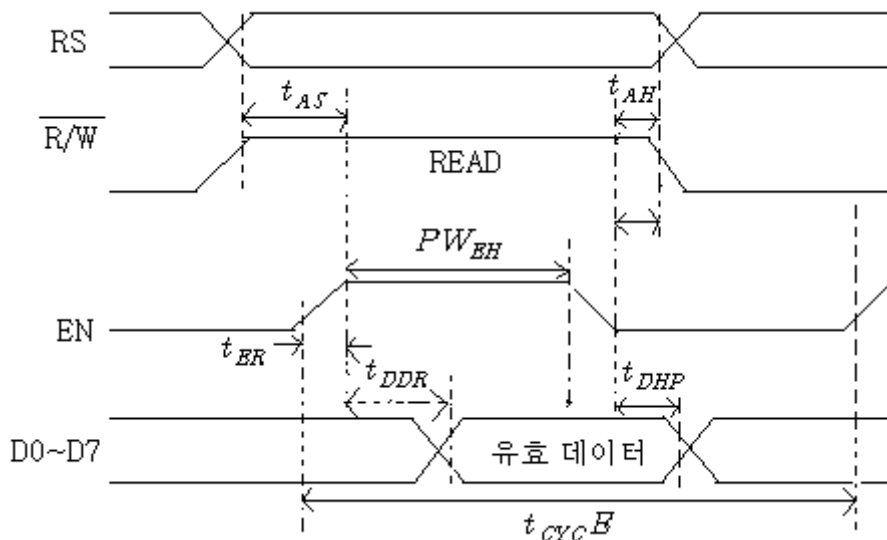


그림 2.21 READ 타이밍도

Fig. 2.21 Time chart of READ

㉠ LCD 회로의 구성

- 레지스터 : LCD controller 내부에는 명령 레지스터(IR)와 데이터 레지스터(DR) 2종류의 8 비트 레지스터가 있으며, 이 레지스터는 RS(Register Select) 신호에 의해 선택된다.
- BF(Busy Flag) : 명령 레지스터의 비트 7번으로 모듈이 다음 명령

을 받을 수 있는지, 없는지를 알려준다. BF = 1 이면 모듈 내부 동작 중 임을 표시하며, 다음 명령을 받을 수 없다. BF = 0 이면 다음 명령을 써 넣을 수 있다.

④ 시리얼 커뮤니케이션 컨트롤러 회로부

설계된 SCC는 CRT와 MODEM간 데이터 교환을 위하여 Z8530 SCC 제어기를 사용하였다. A, B 두 포트는 모두 RS232 비동기식 포트로서 사용하며, 입력 클록은 12.5[MHz]가 입력된다. SCC Rx 인터럽트는 CPU와 wired-or되어 인터럽트시 CPU와 SIO의 SCC 상태 레지스터를 확인하여 각각의 SCC에 대해 인터럽트 처리할 수 있다.

다. 디지털 스피드센서 모듈 (DSSM)

(1) 개 요

DSSM은 다수의 MPU(Magnetic Pickup Unit)로부터 현재의 터빈회전 속도와 속도 변화율 값을 입력받아 각 CPU 모듈의 입력 채널로 신호를 전달하는 기능을 수행한다. 과거의 기계 유압식 조속기는 벨로우즈나 플라이 볼을 이용해 속도를 검출하였으나, 갑작스런 부하 변동 및 응급 상황시 속도 제어 특성이 빠르지 못한 단점이 있었다. 현재는 소자의 발달과 더불어 고도의 정밀성과 빠른 처리 속도를 가진 디지털 방식이 주로 사용되어지고 있으며, 본 연구에서도 실시간 처리 및 응용이 유연한 DSSM을 사용한 디지털 속도 검출 방식을 채용하였다.

DSSM은 2개의 MPU로부터 입력되는 신호 펄스를 분석하고 모니터링 하는 모듈이며, 입력된 신호들은 마이크로프로세서에 의해 속도에 비례하는 값을 CPU로 전달한다. CPU는 실시간적으로 입력된 값을 연산 처리하게 되는데, 이러한 검출 방식은 속도 및 속도 변화율의 판정에 있어 고도의 정밀성을 가능하게 한다. 그림 2.22는 터빈의 회전수를 측정하기 위한 MPU의 부착 위치 및 속도 검출 방식을 도시하였다.

현재 설치된 복제주화력의 터빈 주축에 연결된 메인 기어와 종동축의 기어비, 회전비는 아래와 같다.

기어비 = Main : Slave = 6 : 4

회전비 = Main : Slave = 4 : 6

MPU는 중동축의 기어에 부착되어 회전수를 검출한다. 발전기 및 터빈 주축의 정격 회전수는 3600[rpm]이고, 중동축의 기어 회전수는 5400[rpm]이다.

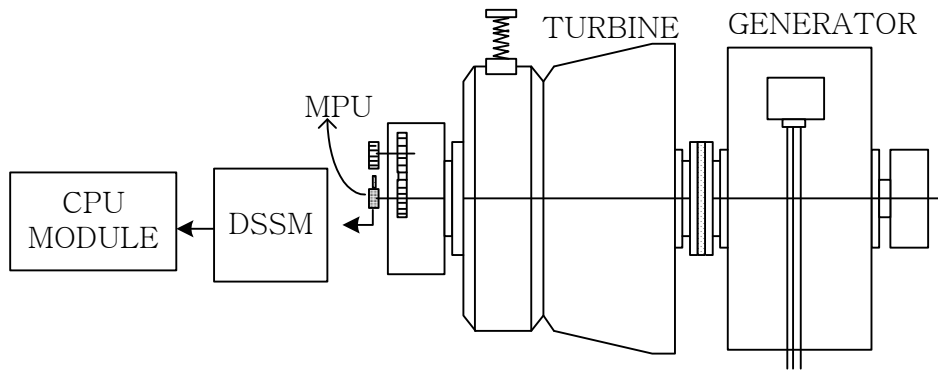


그림 2.22 터빈의 회전속도 측정

Fig 2.22 Measurement of turbine RPM

(2) 적용 발전소 환경 조사

① 덤프(Dump)시 부하별 속도 측정

속도 \ 부하	25[%] (2500[kW])	50[%] (5000[kW])	75[%] (7500[kW])	100[%] (10,000[kW])
최대 회전수 (N_B)	3709 (3.03)	3785 (5.14)	3840 (6.67)	3934 (9.28)
정정 회전수 (N_C)	3665 (1.81)	3703 (2.86)	3713 (3.14)	3788 (5.22)

※1. 부하 차단시 순시 최대속도 상승률은 정격의 111[%]이내여야 함.

2. N_C 는 정격회전수(3600[rpm])이다.

3. 북제주화력발전의 전부하는 10000[kW]이다.

$$\circ \text{ 속도상승율} = \frac{N_B - N_S}{N_S} \times 100[\%] \quad (2.1)$$

$$\circ \text{정정속도 조정율} = \frac{N_c - N_s}{N_s} \times 100[\%] \quad (2.2)$$

② 속도 검출

터빈의 종동축에 연결된 기어의 이(齒)수는 30개이며, 터빈 주축과 종동축의 회전비는 4:6이므로 정격속도(3600[rpm])에서 실제 MPU로 입력되는 회전 속도는 5400[rpm]이다. 이를 감안하여 속도 검출 식을 산정하였다. 아래 표는 속도에 따른 기어 1 이(齒)의 Cycle time을 나타낸다.

표 2.6 속도별 사이클타임

Table 2.6 Cycle time of turbine

주축 RPM	종동축 RPM	주축 C.T.	종동축 C.T.	주축 [rpm]	종동축 [rpm]	주축 C.T. [μ s]	종동축 C.T. [μ s]
1	1.5	2[sec]	1.3[sec]	2600	3900	769.230	512.820
200	300	10.000[ms]	6.667[ms]	2800	4200	714.285	476.190
400	600	5.000[ms]	3.333[ms]	3000	4500	666.666	444.444
600	900	3.333[ms]	2.222[ms]	3200	4800	625.000	410.663
800	1200	2.500[ms]	1.666[ms]	3400	5100	588.235	392.157
1000	1500	2.000[ms]	1.333[ms]	3600	5400	555.556	370.370
1200	1800	1.666[ms]	1.110[ms]	3800	5700	526.315	350.877
1400	2100	1.428[ms]	0.952[ms]	4000	6000	500.000	333.333
1600	2400	1.250[ms]	0.833[ms]	4200	6300	476.190	317.460
1800	2700	1.111[ms]	0.741[ms]	4400	6600	454.545	303.030
2000	3000	1.000[ms]	0.667[ms]	4600	6900	434.782	289.855
2200	3300	909.090 [μ s]	606.060 [μ s]	4800	7200	416.666	277.777
2400	3600	833.333 [μ s]	555.555 [μ s]	5000	7500	400.000	266.667

- 기어 이(齒) 수(N_{teeth}) : 30[개]
- 동기 속도에서의 주기(Cycle time)

$$\text{Cycle Time} = \frac{60}{N_{teeth} \times RPM} [\mu s] \quad (2.3)$$

- Digital speed sensor 모듈의 입력 클럭(f_{osc})으로 25[MHz]
오실레이터 사용
- 정격주파수(f)는 60[Hz] (Synchronize speed)
- 동기 속도(60[rps])에서의 주파수 계산

$$\text{Cycle Time} = \frac{60}{30 \times 5400} = 370.37 [\mu s] \quad (2.4)$$

$$N_{teeth} = C.T. \times f_{osc} = 370.37 [\mu s] \times 25 [MHz] = 9259.25 [\text{개}] \quad (2.5)$$

$$f = \frac{1}{N_{teeth} \times C.T.} = \frac{1}{N_{teeth} \times N_{count} \times T_{count}} = \frac{f_{osc}}{N_{teeth} \times N_{count}} = \frac{25 \times 10^6}{30 \times N_{count}} \quad (2.6)$$

- 1[rps]일 때의 N_{count}

$$N_{count} = \frac{f_{osc}}{N_{teeth} \times f} = \frac{25 [MHz]}{30 \times 1.5} = 555,555.6 [\text{개}]$$

$2^X = 555,555.6$ 이므로 약 20bit의 해상도가 요구된다.

본 연구에서는 계수기의 여유를 생각하여 24[bit] 카운터로 설계하였다.

(3) 구 성

DSSM은 두 개 이상의 MPU로부터 입력된 신호 펄스를 분석하고 모니터링하기 위한 메인시스템의 subsystem으로 사용된다. DSSM은 Input buffer, Select logic, Divider, One shot, Counter, Latch부로 크게 나눌 수 있다. 터빈의 회전수로부터 검출된 MPU 출력 신호는 비교적 저주파수 대역이므로 신호 속의 고주파 성분을 제거하기 위해 지역 통과 필터를 첨가한다. 필터를 거친 입력 신호는 DSSM의 디지털 필터링 알고리즘에 의해 연산되어진다. 입력 신호의 주기는 점퍼를 통하여 하드웨어적으로 조정이 가능하도록 고려하였다.

필터링 알고리즘은 여러 종류가 있으나, 여기서는 일정 수의 샘플링 값

을 평균하여 정밀도를 증가시키는 방법을 택하였다. 또한 각 샘플링 구간의 속도 미분값을 소프트웨어 방식으로 처리하여 일정값 이상의 변화율을 가지는 값은 노이즈로 처리함과 동시에 자체 진단의 기능을 추가하였다.

본 연구에서 채택하고 있는 속도 검출 방식은 기어 이(齒) 하나가 MPU의 위치를 통과하는 순간 단안정 펄스회로에 의해 카운트를 시작하여 다음 이(齒)가 오면 카운터를 멈추고 현재의 값을 래치한다. 래치와 동시에 CPU에 인터럽트를 걸어 명령이 수행되고, 카운터한 값을 CPU 레지스터에 저장한 후 연산에 의해 실제의 회전수를 구한다.

① 블록 다이어그램

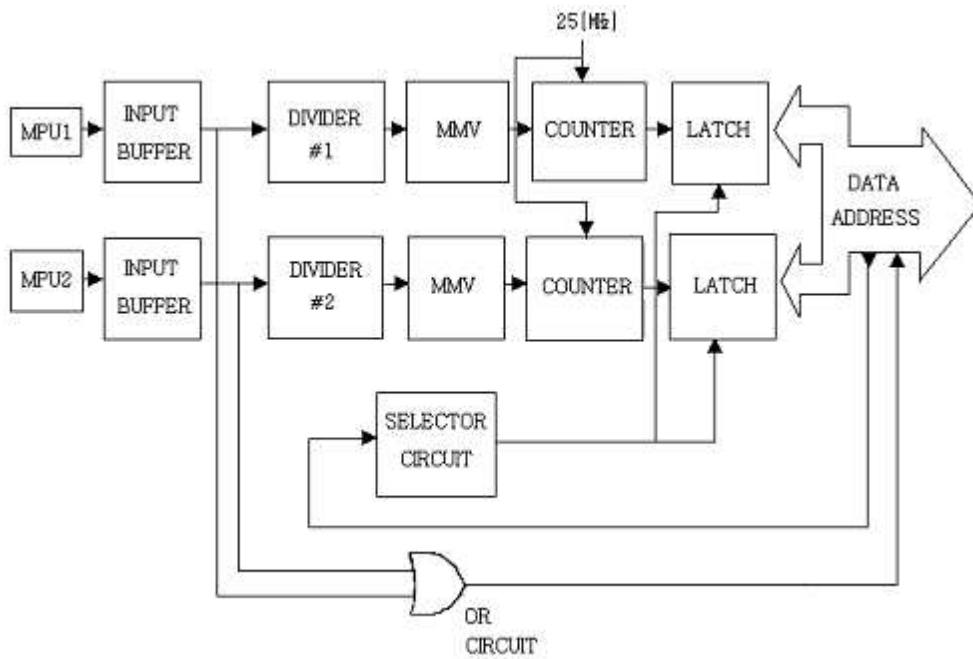


그림 2.23 DSSM 블록 다이어그램
Fig. 2.23 Block diagram of DSSM

② 스피드센서

속도 검출에 사용되는 장치는 주로 MPU와 근접 스위치이다. 두 장치의 특성은 다음과 같다.

- Input frequency range
 - MPU : 50[Hz] ~ 25[Hz]
 - Prox. : 0.04[Hz] ~ 2[kHz]
- Input amplitude
 - MPU : 1[V_{rms}] min , 25[V_{rms}] max , Freq > 20
 - Prox. : 10[mA]
- Input impedance
 - MPU : 2,000[Ω]
 - Prox. : 2,000[Ω]
- Isolation voltage
 - MPU : 500[V_{rms}]
 - Prox. : None
- Resolution : 0.0015[%] of range/LSB
- Speed accuracy (max) : 0.01[%] over temperature range
- Derivative accuracy (max) : 0.1[%] of range (p-p)
- Speed filter : 5 ~ 10,000[ms] (2 real poles)
- Derivative filter : 5 ~ 10,000[ms] (1 pole + speed filter)
- Acceleration limit : 1 ~ 10,000[%]/[s]

MPU는 고속의 회전수 검출용으로 근접 스위치는 저속의 회전수 검출용으로 주로 사용되고 있다. MPU는 픽업의 앞부분에 자기장이 형성되는데, 그 속으로 금속성 물질이 접근하면 전압을 발생시킨다.

MPU의 출력 전압은 다음의 3가지 조건에 영향을 받는다.

- 검출된 금속 물질의 표면 속도가 증가할수록 출력 전압 증가

- 마그네틱 픽업과 기어 이(齒) 사이의 간극이 클수록 출력 전압은 감소
- 출력 전압 파형은 마그네틱 픽업의 위치와 기어의 크기 및 형태에 의해 결정

아래의 식에 의해 표면 속도, 직경 피치, 기어 모듈을 구할 수 있다
[15],[16],[17]

$$\text{표면속도} = \frac{RPM \times \pi \times \text{직경}}{60} \quad (2.7)$$

$$\text{직경피치}(D_p) = \frac{Z \text{ 이(齒) 수} + 2}{\text{기어직경}(D.P)} \quad (2.8)$$

$$M(\text{기어모듈}) = \frac{\text{기어직경}(D.P)}{Z \text{ 이(齒) 수} + 2} \quad (2.9)$$

③ 로직 선택 및 구동 회로부

DSSM 은 2채널로 구성되어 있으며 Select logic은 아래의 어드레스맵에 의해 선택되어진다.

표 2.7 로직선택과 어드레스맵

Table 2.7 Select logic and address map

Address	유효 데이터 영역	액세스 단	Assignment
Base + 0000	D24~D0	Long Word	Channel #0
Base + 1000	D24~D0	Long Word	Channel #1
*Base = 05410000			

기어 이(齒), 한 주기의 카운터가 끝남과 동시에 인터럽트 신호를 CPU로 보낸다. 이때 2개의 채널이 각 유효 데이터를 알리는 상태비트가 있어 인터럽트 신호가 걸리면 CPU는 상태비트를 읽어 유효 데이터 및 채널을 선택한다. 속도 검출에 있어 한 개의 이(齒)에 한 번씩 검출하는 것이 이상적이지만 기어의 이(齒)개수에 의한 카운터의 계수 범위 및 샘플링 시

간을 점퍼로 조정 가능하도록 구성하였다.

채널 선택을 위한 PAL 프로그램은 부록의 6을 참조한다.

④ 단안정 회로부

㉠ 단안정 펄스

그림 2.24는 채널 당 3개의 PLD소자를 이용한 카운트 제어 신호 생성 로직이다.

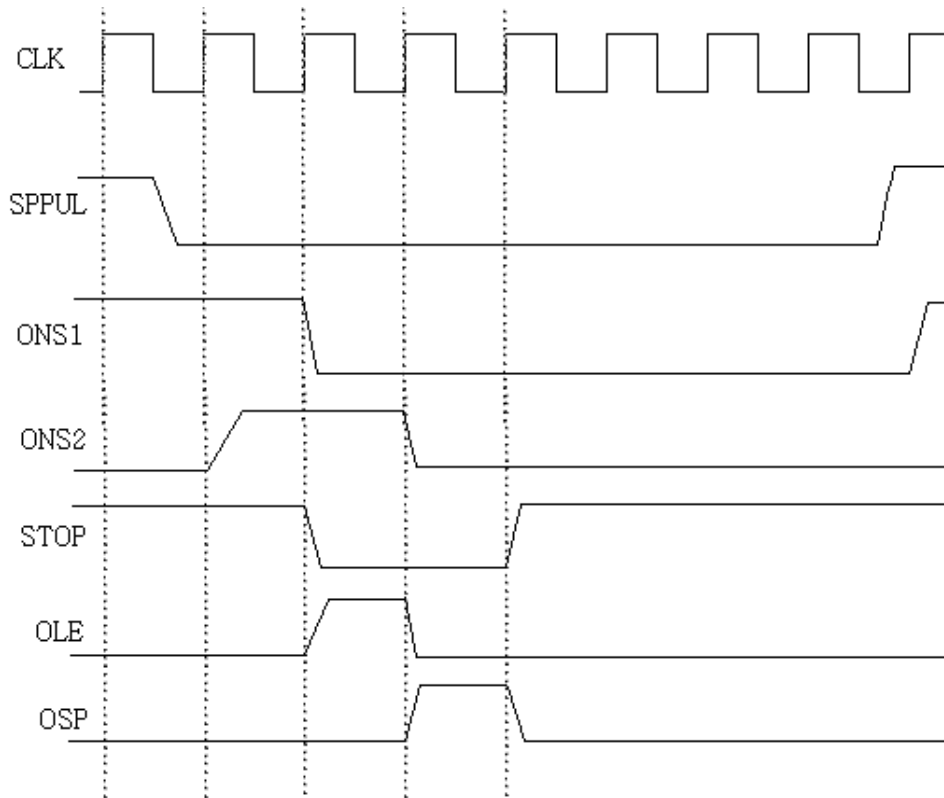


그림 2.24 카운터 초기화 타이밍 선도

Fig. 2.24 Time chart of counter for initializing

입력 CLK으로 25[Mhz]의 오실레이터를 사용하며, 1사이클 타임은 40[ns]이다 MMV(Monostable Multivibrator)는 MPU로부터 입력된 신호가 외부의 R, C 값에 의해 일정 시간 "L" 로 가도록 하여 STOP, OLE 및 OSP의 신호를 조합 생성하였다.

㉞ Stop

MMV와 ONS1, ONS2를 조합하여 Stop 신호를 만들 수 있다.

ONS1 : =MMV # !ONS2 & ONS1;

ONS2 : =ONS1 & MMV;

STOP : =!ONS2 & !FULL;

Stop 출력이 "H"일 때만 카운터가 동작을 하고 "L"일 때 계수동작은 정지함과 동시에 현재의 데이터 값을 래치 시키고 CPU에 인터럽터 (IRQ#5)를 걸어 예외처리루틴을 수행한다.

㉟ OLE(Output Latch Enable) 및 OSP(Output Synch Preset)

인터럽트 명령 사이클이 끝나면 OSP를 "H"로 어서트하여 카운터를 초기화시키고 다시 카운트 동작을 수행한다. OLE는 "H"일때 입력에 따른 출력값이 변하고 "L"이면 현재의 상태 값을 저장한다. CPU는 래치의 OC(Output Control) 단자를 "L"로 어서트하여 저장된 데이터 값을 가진다.

OLE := STOP & ONS2 & ONS1;

OSP := OLE # FULL;

FULL은 24[bit] 카운터의 오버플로를 검출하여 만약 세트되면 조합에 의해 카운터를 초기화시킨다.

㉔ 측정결과

그림 2.25는 실제로 PLD 소자를 프로그램하여 구성된 회로를 로직 에널라이저로 측정한 결과값이다.

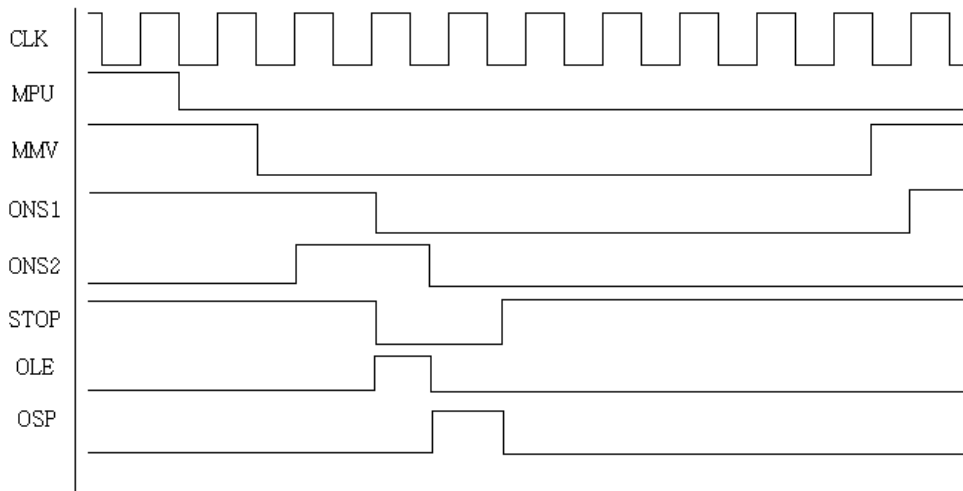


그림 2.25 스피드센서 PAL 측정파형

Fig. 2.25 Waveform of Speed sensor PAL

⑤ 카운터 회로부

㉕ 카운터

Digitizer speed sensor 모듈의 카운터는 24[bit]로 구성되어져 있다 . 1 RPS로 회전할 경우 25[MHz] 의 클록으로 셀 수 있는 최대수는

$$N_{count} = \frac{f_{osc}}{N_{teeth} \cdot f} = \frac{25[MHz]}{30 \cdot 1.5} = 555555[개] \quad (2.6)$$

카운터는 24[bit]로 설계하였으며 현장 조건에 따라 prescaler로 입력신호를 분주하여 샘플링 주기를 하드웨어적으로 가변 할 수 있도록 설계하였다. 이진동기 카운터 방식으로는 크게 두 가지가 널리 사용되고 있는데, 직

렬 캐리 동기 카운터와 병렬 캐리 동기 카운터가 있다. 본 연구에서는 직렬 캐리방식을 채택하였으며 기본적인 계수방식은 그림2.26과 같다.

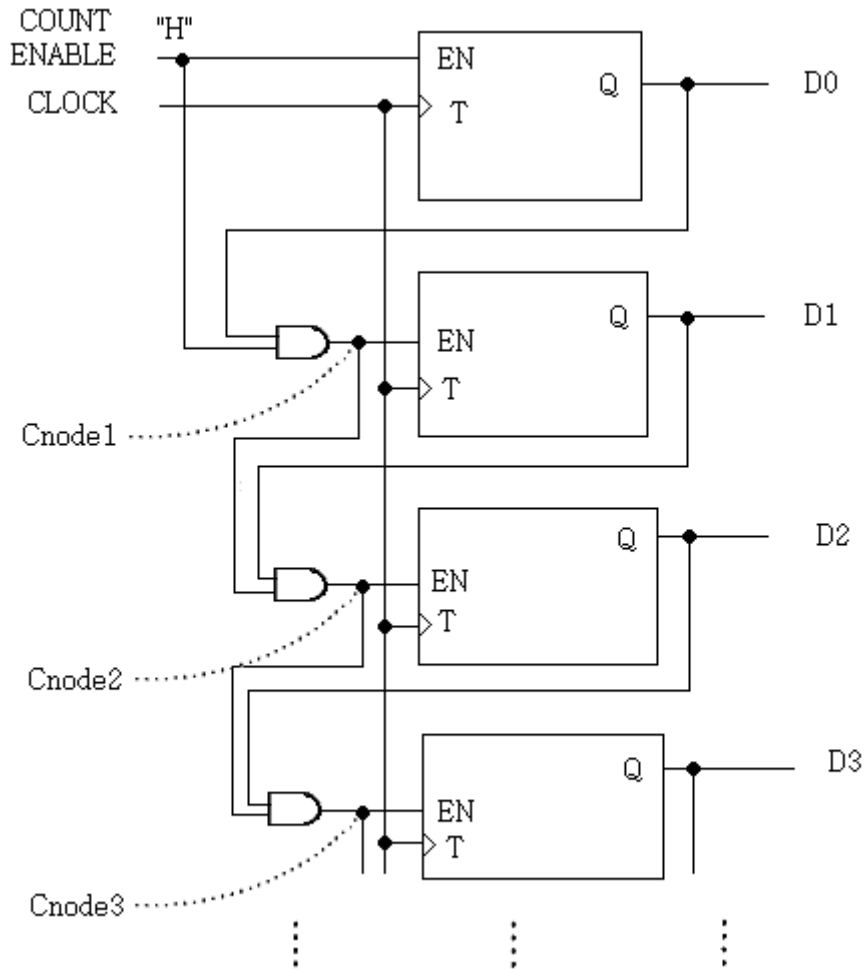


그림 2.26 2진 동기카운터

Fig. 2.26 Synchronous binary counter

㉠ 카운터의 동작 타이밍

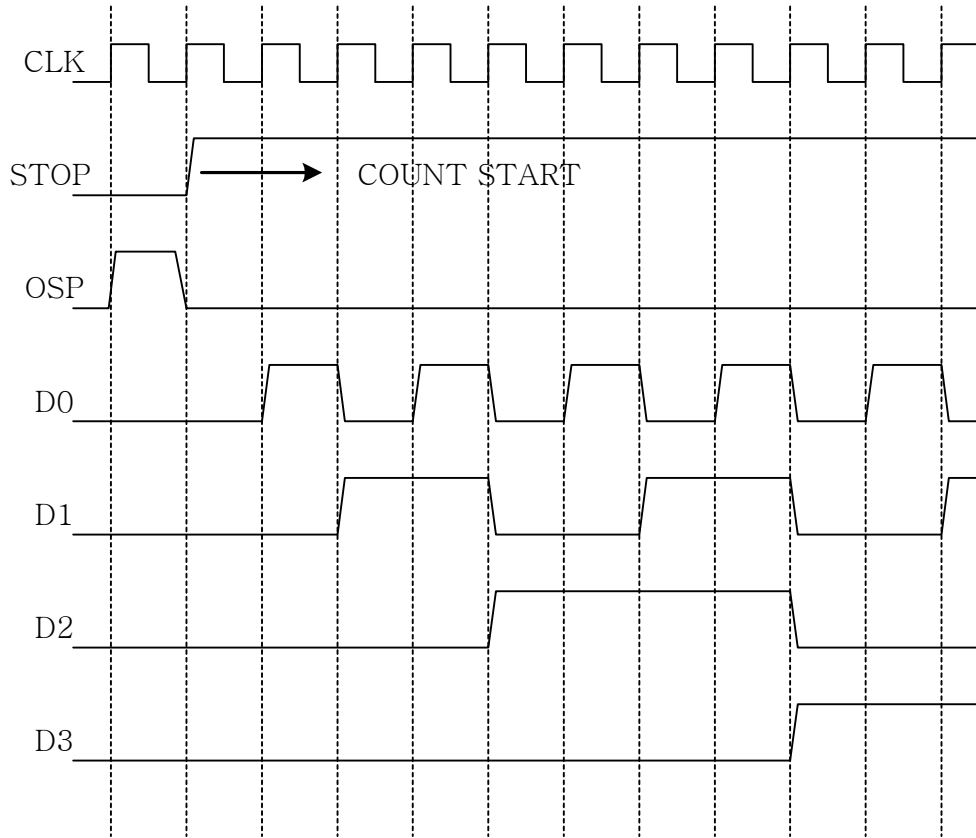


그림 2.27 2진 동기카운터의 타이밍선도

Fig. 2.27 Time chart of synchronous binary counter
(Serial carry type)

㉔ 카운터 PLD 프로그램은 부록의 7을 참조한다.

㉕ 측정결과

그림 2.28은 실제로 PLD 소자를 프로그램하여 구성한 카운터 회로를 로직 에널라이저로 측정한 결과값이다.

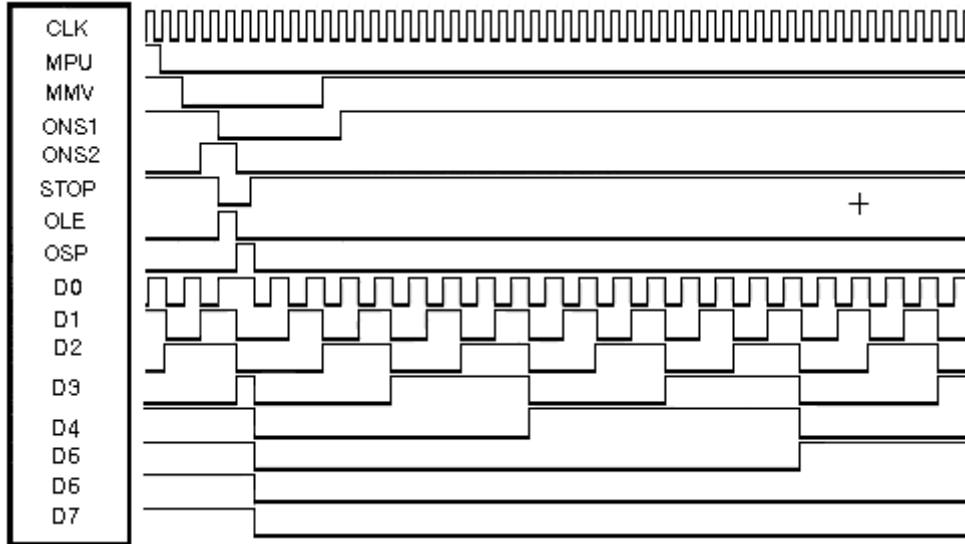


그림 2.28 스피드센서 카운터회로 측정 파형

Fig. 2.28 Waveform of speed sensor counter circuit

라. 아날로그 출력모듈

(1) 개요

이 모듈은 CPU로부터 디지털 신호를 받아, 외부기기를 위하여 4~20 [mA]의 구동 전류신호를 발생시킨다. CPU는 이 모듈에 디지털 신호를 보내고, 또한 제어하고 있는 모듈의 출력 중 디지털 데이터에 의해 영향을 받는 출력의 어드레스 정보를 보낸다.

DAC(Digital to Analog Converter)는 디지털 값을 latch 한 뒤 아날로그 신호로 변환시키고, 그 채널에 대한 driver 상의 신호를 보낸다.

아날로그 신호는 전류 신호가 직접 출력되고 이것이 4~20[mA] 전류를 흐르도록 하는 트랜지스터를 구동한다. TR에서 출력된 전류 신호는 0.01[%] 243[Ω]의 정밀 저항을 거쳐 공급된다.

DAC는 reset일 때나 데이터가 000H일 때 4[mA], 3FFH 일 때 20[mA]

를 유지 하도록 하였다. CPU는 2 out of 3 voting 방식을 채택하므로 이 모듈의 모든 출력을 사용할 수 없게 하는 I/O lock 신호를 발생시킬 수 있다. 만약 CPU가 자동진단 테스트중 이 모듈의 문제를 감지하면 CPU는 Fault LED를 점등시킬 것이다. 특징으로는 12[bit] 분해능, 8개의 정전류 출력 (4~20[mA]), 2 out of 3방식의 시스템을 위한 I/O lock 회로로 되어있다. 아날로그 출력모듈 어드레스 선택은 표 2.8과 같다.

표 2.8 아날로그 출력모듈 어드레스 선택

Table 2.8 Analog output module address selection

Address	유효 데이터 영역	액세스 단	Assignment
BASE + 0000	D 11 ~ D 0	LONG WORD	CHANNEL #0
BASE + 1000	"	"	CHANNEL #1
BASE + 2000	"	"	CHANNEL #2
BASE + 3000	"	"	CHANNEL #3
BASE + 4000	"	"	CHANNEL #4
BASE + 5000	"	"	CHANNEL #5
BASE + 6000	"	"	CHANNEL #6
BASE + 7000	"	"	CHANNEL #7

* BASE = 05420000

(2) 회로 설명

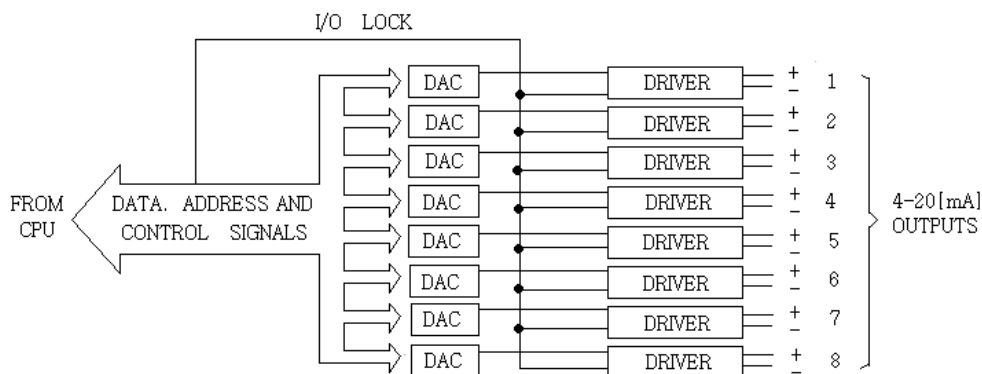


그림 2.29 아날로그 출력모듈 블록도

Fig. 2.29 Analog output module block diagram

① 인터페이스 회로부 : 디코더, 채널 셀렉터

Interface 회로부는 CPU에서 전달되는 데이터를 디코더와 Channel selector 회로에 의해 선택되어지는 채널에 데이터를 전송한다. 디코더 회로는 GAL16V10을 이용하여 모듈에 할당된 base 어드레스를 디코더하여 액티브 되도록 하였다.

CPU의 데이터는 74F245에 나누어서 전송되어, D/A 컨버터와 PLD 소자에 각각 전송된다.

74F245는 데이터 트랜시버로써 각각 I/O locked signal과 R/W signal의 제어에 따라 동작한다. PLD 소자에는 CPU 데이터의 자동 진단 테스트를 위한 데이터가 전송된다.

어드레스와 I/O Locked Signal은 또 하나의 PLD로 전달되어 74F138을 active 시키는 종작을 하게 되면 138은 어드레스 LSB와 PLD의 액티브 신호를 받아 8개의 D/A 컨버터중 하나를 선택하는 채널 selector 역할을 하게 된다.

모듈 선택 PLD 프로그램은 부록의 8을 참조한다.

② D/A 컨버터 회로부

어드레스 디코더에 의해서 D/A 컨버터가 active되면 12비트의 데이터가 입력단에 래치되고, 그에 따른 아날로그 값이 출력된다.

MP1208은 12비트 입력, 두 개의 전류 출력을 가진 D/A 컨버터로 마이크로프로세서로 데이터 버스와 8/4 비트 래치 입력 interfacing을 한다.

MP1208 D/A converter의 Timing chart는 그림 2.30과 같다.

D/A 회로의 라이트 타이밍은 \overline{CS} 가 어서트된 후 데이터 Setup time은 최소 100[ns]이며 \overline{WR} 이 니게이트된 후 데이터 홀드 타임이 최소 90[ns]로 동작한다.

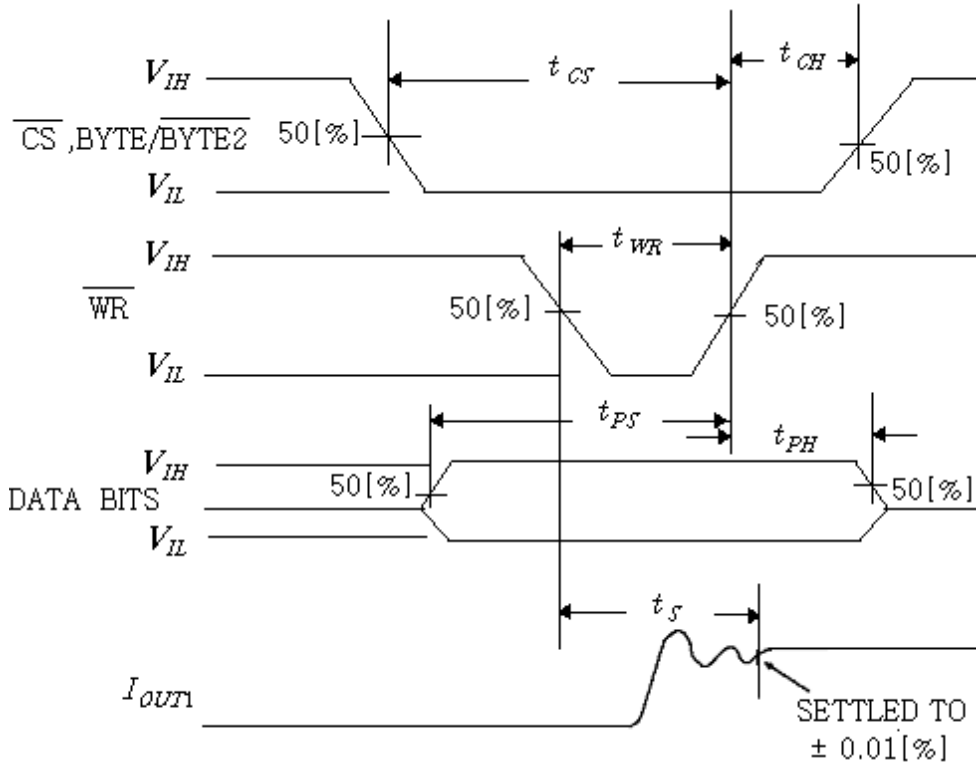


그림 2.30 D/A 컨버터의 타이밍도

Fig. 2.30 Time chart of D/A converter

③ I/O locked 회로부

이 모듈은 모든 아날로그 출력을 사용하지 못하도록 모든 D/A 컨버터의 출력단에 I/O locked 출력을 입력 받을 수 있도록 설계하였다. I/O locked 신호가 액티브되면 비교기회로에 의해 부(-)로 포화되어 전류구동회로의 트랜지스터를 OFF 하도록 하였다.

④ 전류 구동 회로부

Current driver 회로는 이득조정회로와 Bias adjust회로, 전류변환 회로로 구성 되어있다. 이득조정회로는 D/A 컨버터의 전류 출력을 전압으로 변

환하며 이득조정용 가변저항 VR_1 에 의해 0~10[V]까지 조정 가능하다.

이 출력값은 bias조정용 가변저항 VR_2 에 의해 제공되는 전류 4[mA]와 가산하여 전류변환 회로로 유입된다. 전류변환 회로는 출력 임피던스 변화에 변동 없이 일정한 전류를 출력하도록 설계되어 있으며 1N6284A Zenor diode에 의해 출력이 발생하는 과전압 및 부(-) 출력을 막아준다.

식(2.10)에서 R_L 이 변하더라도 출력은 정전류로 설계되어 있으므로, 전류 오차를 일으키지 않는다. 신호의 크기를 4[mA]~20[mA]로 변환해서 전송하나 필드가 단선된 경우는 전류가 0이 되므로 경보의 검지도 가능하게 된다.

$$V_{out} = \frac{R_L + R}{R} \times V_{in} \quad (R_L = \text{부하저항}) \quad (2.10)$$

입력 전압이 0[V] 일 때 4[mA]의 전류가 흐르도록 VR_2 를 세트하므로 OP앰프 TL082a에는 옵셋조정이 설정되어 있지 않다. 다음에 10[V]의 입력을 부여하고 TL082a출력이 -1.6[V]가 되도록 VR_1 을 세트하나 정전류 출력회로의 오차를 포함시켜서 조정하기 때문에 출력 전류가 20[mA]가 되도록 맞추어야 한다.

※ 부하에는 긴 케이블이 접속되어 있는 경우가 많고 본회로 출력단의 발진을 방지하기 위해 OP 앰프 TL082b의 궤환 저항과 병렬로 0.01[μ F]의 컨덴서를 삽입한다. 삽입한 것으로 인해서 고주파 응답이 나빠지나 용도를 생각하면 특별한 문제는 없다.

⑤ 전압 제어 회로부

LH0070는 온도 변화와 단락회로에 영향을 받지 않는 전압제어기로 10[V] \pm 0.02[%]이다.

제 3 장 소프트웨어 설계

3.1 개 요

인텔리전트 디지털 조속기 소프트웨어는 운영 프로그램과 조속기 제어에 필요한 응용 프로그램으로 구분된다.

CPU 기동에 필요한 운영 프로그램은 전원인가 후 하드웨어 리셋에 의해 CPU내의 프로그램 카운터 초기화나 벡터 테이블을 정립하고 CPU 레지스터, 메모리, I/O등의 이상 유무를 테스트하며, 시스템에 사용되는 모든 슬레이브 디바이스를 초기화한다.

응용 프로그램은 조속기제어에 필요한 입출력 프로세싱, governing, PID, Inter communication and voting, 시스템 shutdown 프로그램 등을 수행한다. 또한 2 out of 3 voting을 수행하기 위해 "A" CPU와 "C" CPU 에 입력되는 모든 값은 Inter communication 프로그램에 의해 데이터를 교환할 수 있도록 하여, 3개의 모든 CPU가 입력된 데이터를 공유하도록 하였다. 각 CPU는 응용 프로그램에 의해 데이터를 처리하고 처리 결과를 재 교환하여 불량 데이터를 차단 할 수 있도록 하였다.

3.2 시스템 프로그램 구성

시스템 프로그램 구현은 그림 3.1과 같이 메인 프로그램, 초기화 프로그램, 셀프테스트 프로그램 및 인터럽트 프로그램으로 이루어졌다.

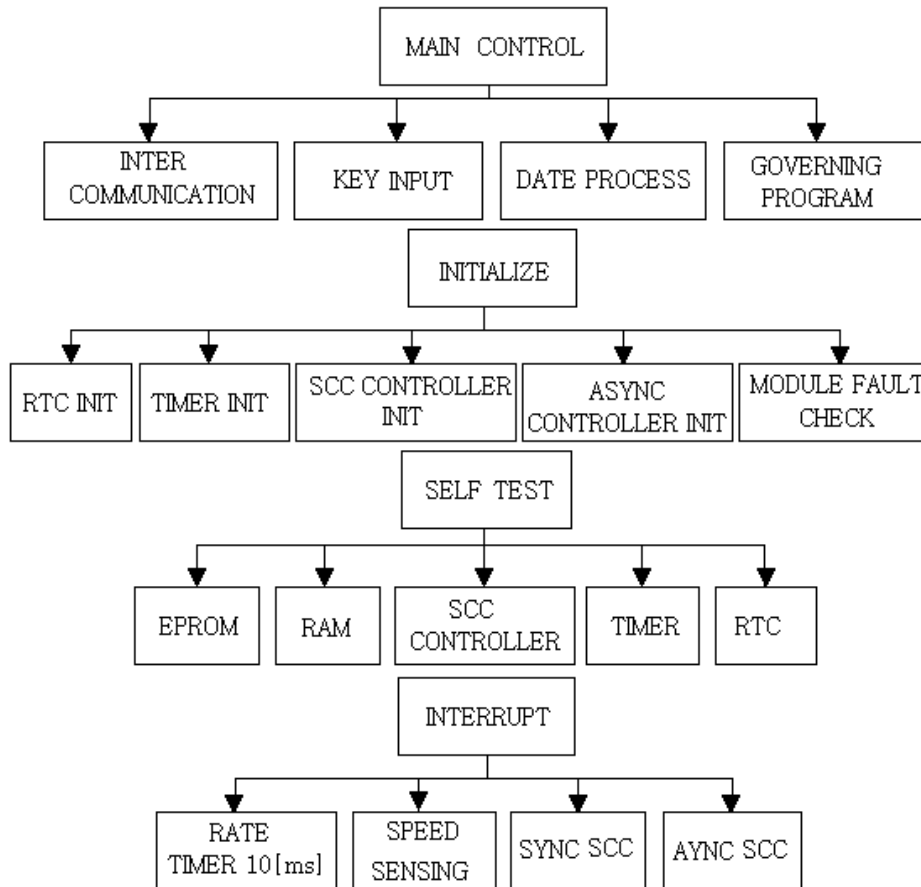


그림 3.1 시스템 프로그램 구성도

Fig. 3.1 Structure of system program

그림 3.2는 프로그램 Rate group timing도를 나타낸 것으로 조속기 제어(응용) 프로그램은 각 작업 별로 Rate group을 정하여 타이머에 의해 매 10[ms]마다 주기적으로 그림 3.1 메인프로그램을 수행하도록 하였다.

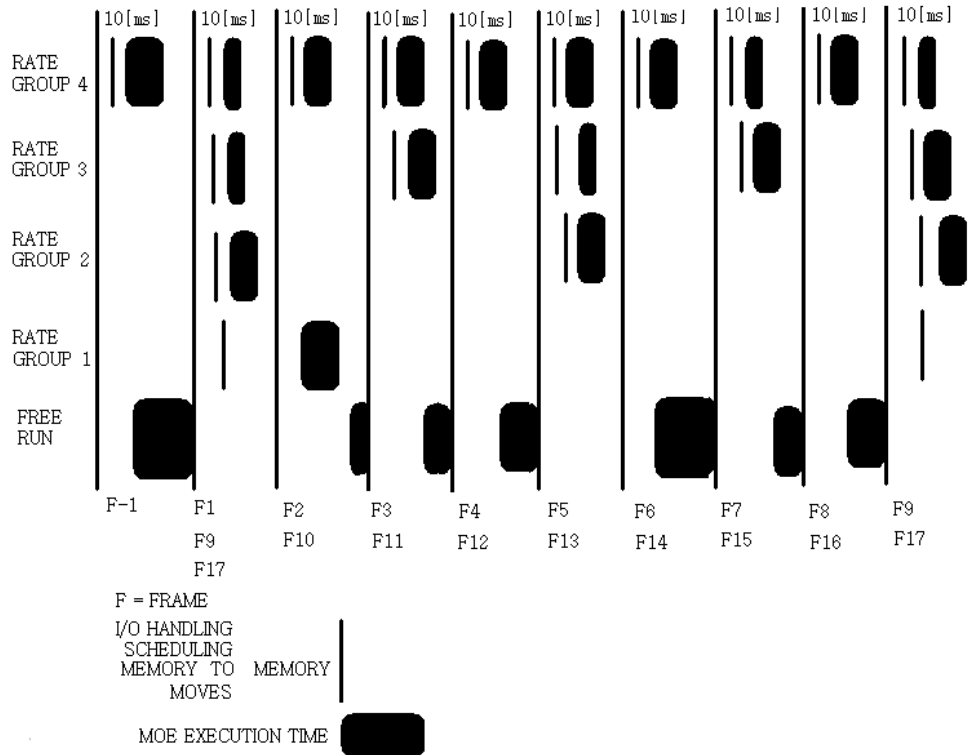


그림 3.2 프로그램 RATE GROUP TIMING

Fig. 3.2 Timing chart of program rate group

3.3 프로그램 개발 내용

3.3.1 SELF - TEST 프로그램

가. CPU TEST 프로그램

CPU의 모든 레지스터에 데이터를 리드한 후 라이트 하여 CPU 레지스터의 이상 유무를 확인한다.

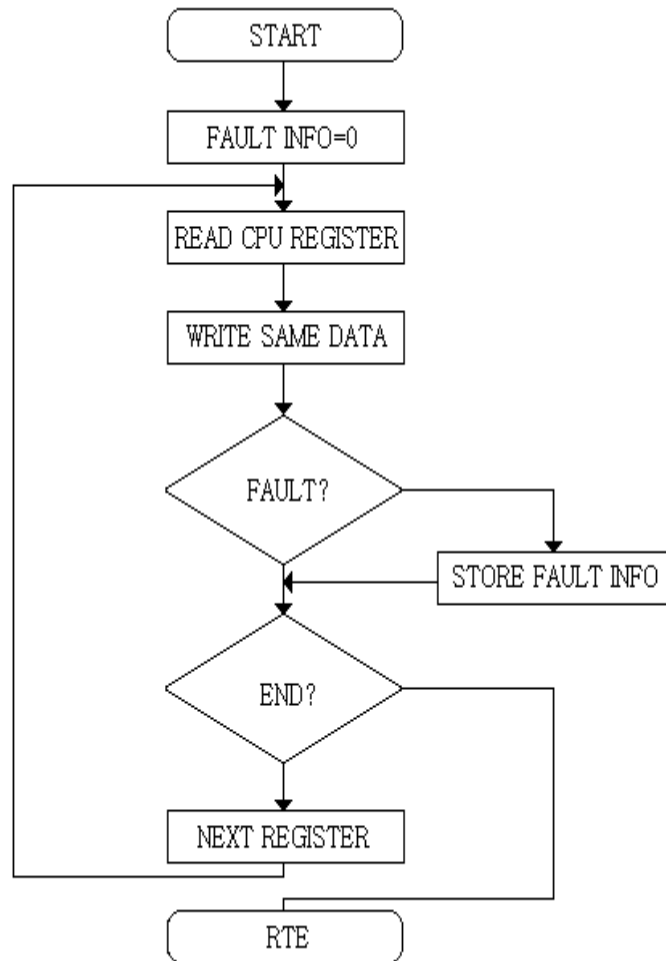


그림 3.3 CPU 테스트 프로그램 흐름도

Fig. 3.3 Flow chart of CPU test program

나. 주 메모리 테스트 프로그램(데이터용 SRAM)

데이터용 SRAM영역(2000000 ~ 21FFFFC)에 액세스 단위를 룽워드
로 데이터를 리드한 후 같은 번지에 리드한 데이터를 라이트 하여 리드/
라이트된 데이터를 비교함으로써 메모리 영역의 이상 유무를 확인한다.

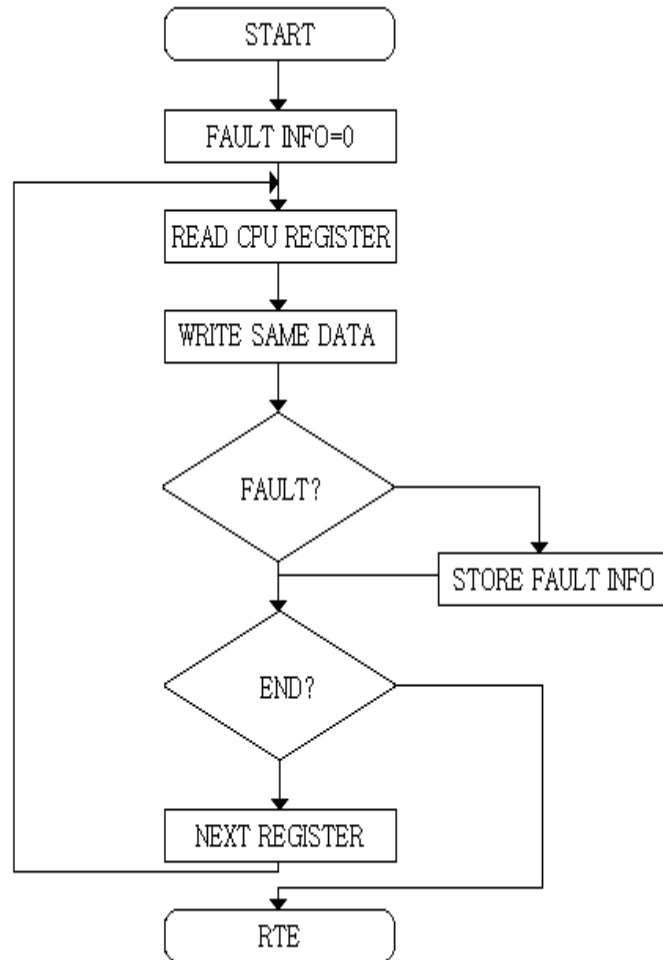


그림 3.4 데이터용 SRAM의 흐름도

Fig. 3.4 Flow chart of SRAM for data

다. 주 메모리 테스트

명령용 SRAM영역(3000000 ~ 31FFFFC)에 액세스 단위를 통위드로 0과 1의 데이터를 라이트하고 리드하여 명령 메모리 영역의 이상 유무를 확인한다.

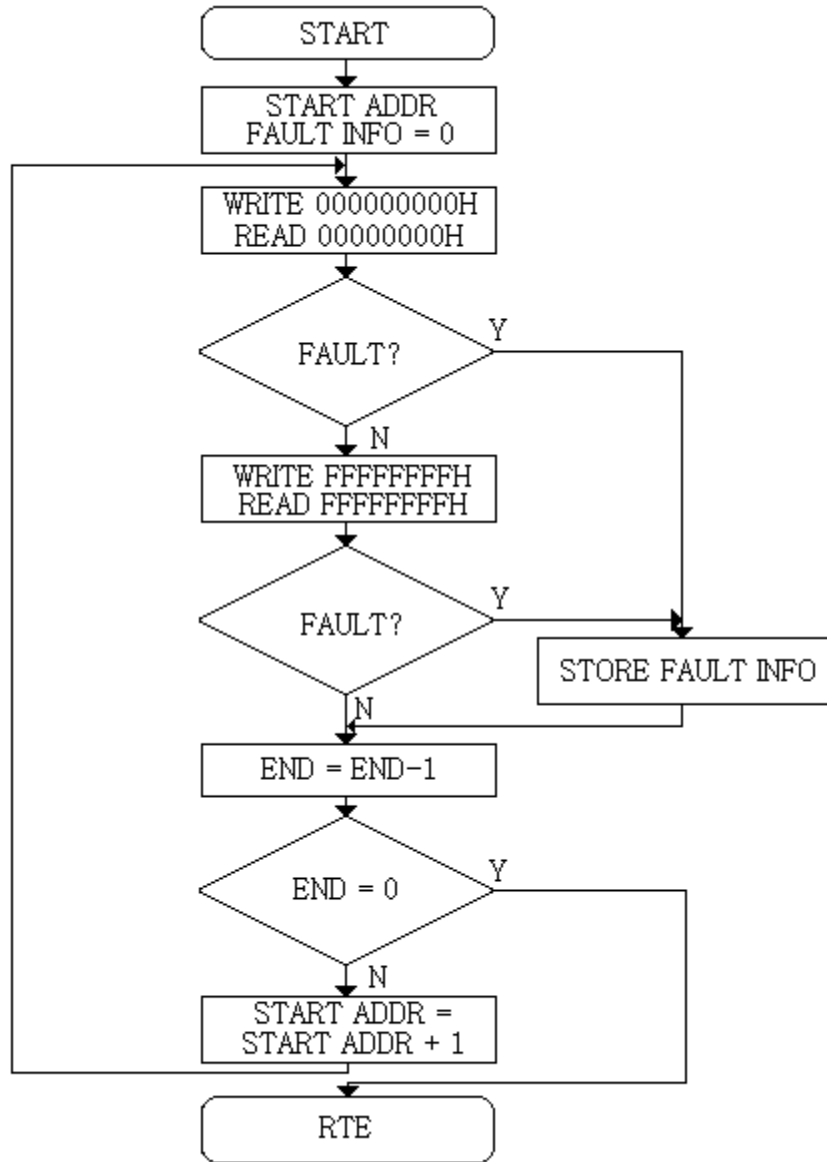


그림 3.5 명령용 SRAM의 흐름도

Fig. 3.5 Flow chart of SRAM for instruction

라. ROM 테스트 프로그램

ROM 메모리 영역(00000000 ~ 00FFFFFC)의 각 어드레스에 데이터를

모두 합산한 후 ROM의 체크섬 값과 비교하여 ROM 메모리 영역의 이상 유무를 확인한다.

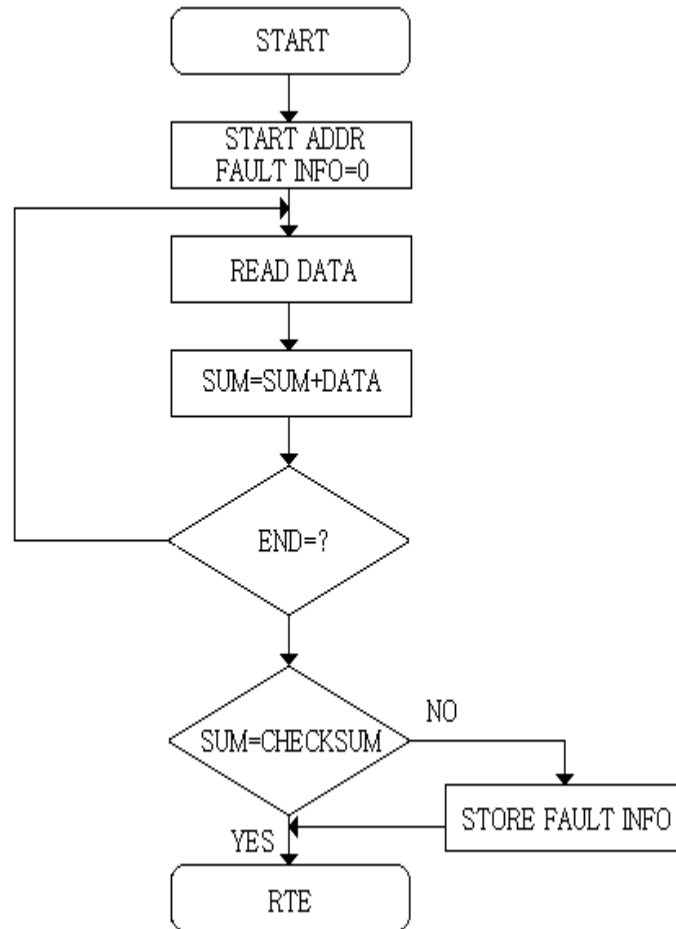


그림 3.6 ROM 테스트의 흐름도

Fig. 3.6 Flow chart of ROM for test

마. 백업용 SRAM 테스트 프로그램

백업용 SRAM 메모리 영역(4000000 ~ 403FFFC)의 모든 데이터를 합산한 후 SRAM내의 체크섬과 비교하여 체크섬 테스트를 한 후, 데이터 액세스

단위를 롬워드르하여 데이터를 리드하고 같은 번지에 리드한 데이터를 라이트 하여 리드/라이트된 데이터를 비교함으로써 메모리 영역의 이상 유무를 확인한다.

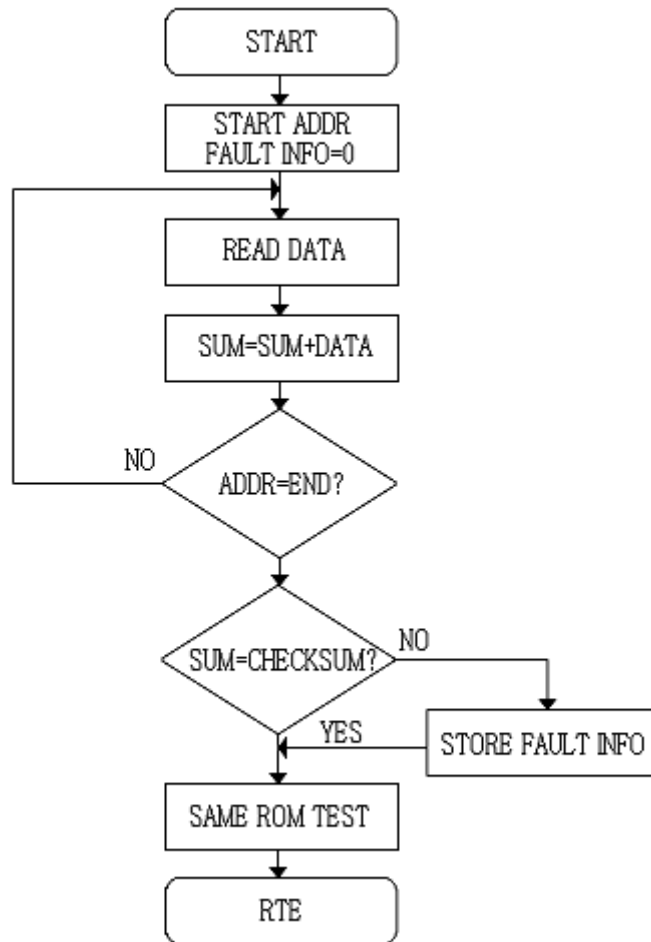


그림 3.7 백업용 SRAM의 흐름도

Fig. 3.7 Flow chart of SRAM for backup

바. RATE 타이머 테스트 프로그램

rate 타이머를 Rate generator 모드로 설정하고 일정시간 지연 후에

타이머 레지스터를 읽어 설정값과 비교하여 타이머의 이상 유무를 확인한다.

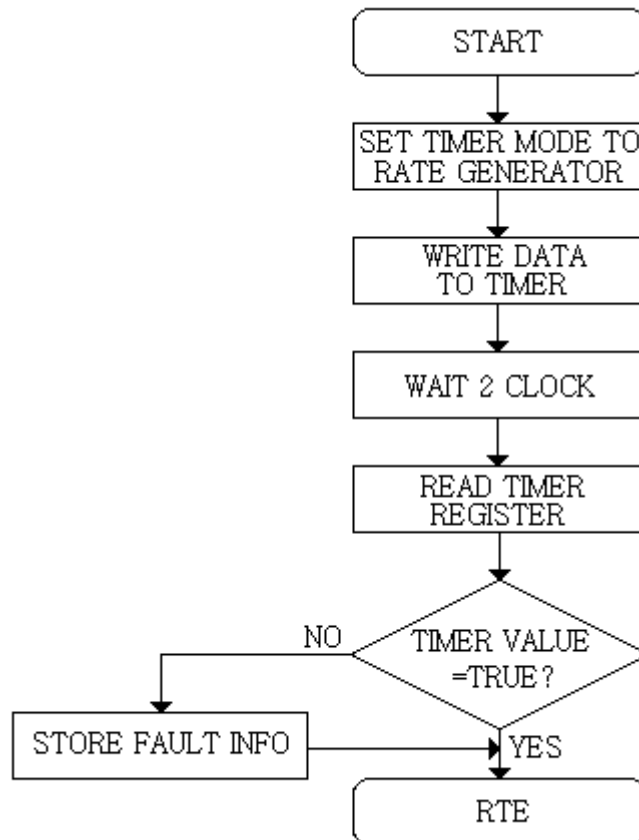


그림 3.8 RATE 타이머 흐름도

Fig. 3.8 Flow chart of SRAM for data

사. 리얼 타임 클록 테스트 프로그램

RTC의 내부 배터리 체크 및 메모리테스트로 RTC 메모리 번지의 데이터를 리드한 후 같은 번지에 리드한 데이터를 라이트 하여 리드/라이트된 데이터를 비교함으로써 메모리 영역의 이상 유무를 확인한다. 최초 데이터 리드/라이트시 메모리 fault이면 Battery bad로 판단한다.

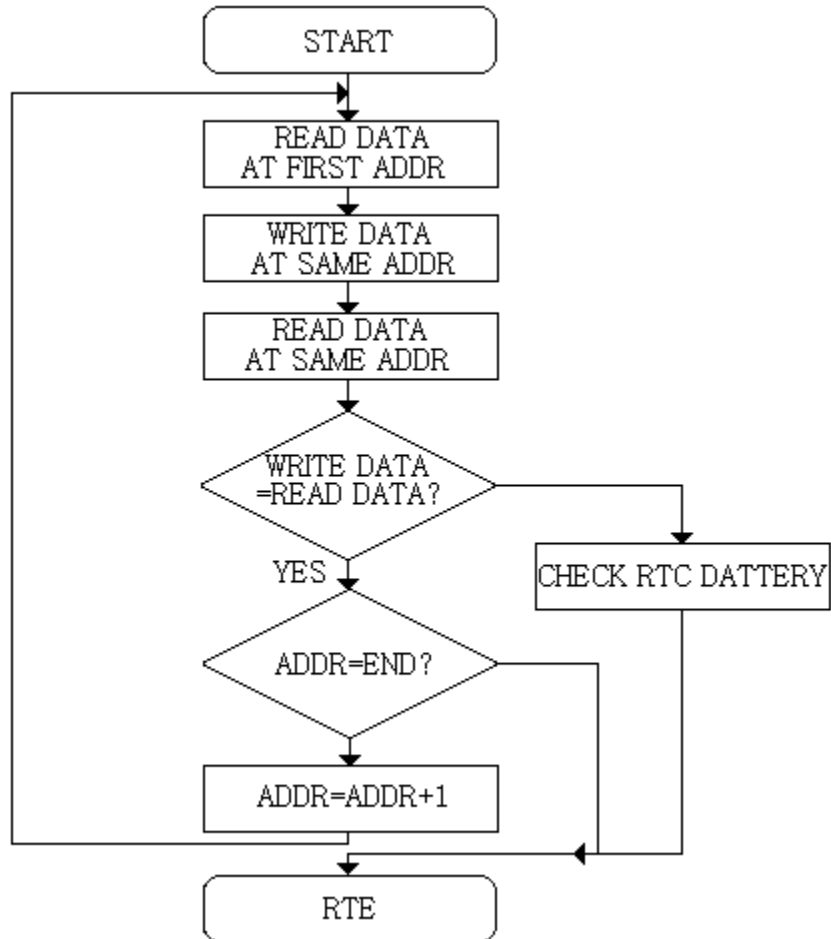


그림 3.9 리얼타임 클럭 흐름도

Fig. 3.8 Flow chart of real time clock

아. 시리얼 커뮤니케이션 컨트롤러 테스트 프로그램

SCC는 루프 백 모드를 이용하여 테스트할 수 있으며, 트랜스퍼 단자에 데이터 0x7E를 전송하여 리시버 단자로 0x7E를 루프백하여 두 데이터를 비교함으로써 테스트 할 수 있다.

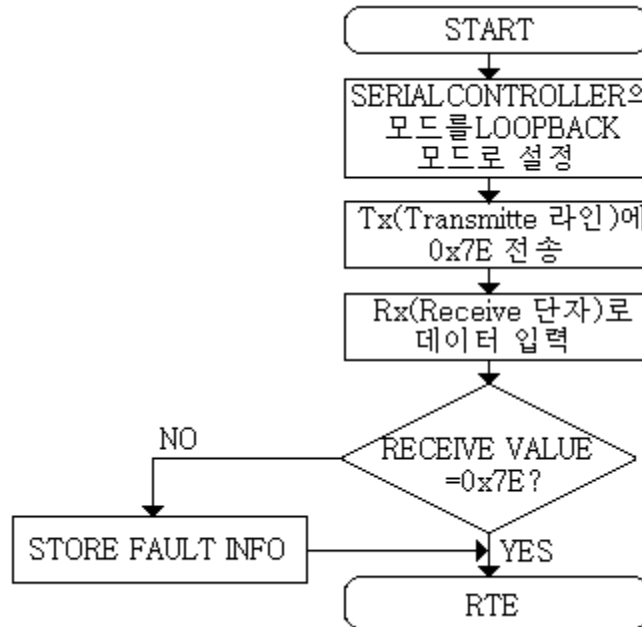


그림 3.10 시리얼 커뮤니케이션 컨트롤러 흐름도

Fig. 3.10 Flow chart of serial communication controller

3.3.2 초기화 프로그램

가. 부트스트랩(Bootstrap) 프로그램

명령 실행 속도를 향상시키기 위해 ROM 영역의 코드를 고속의 주 메모리인 명령용 SRAM영역으로 옮기는 프로그램이다.

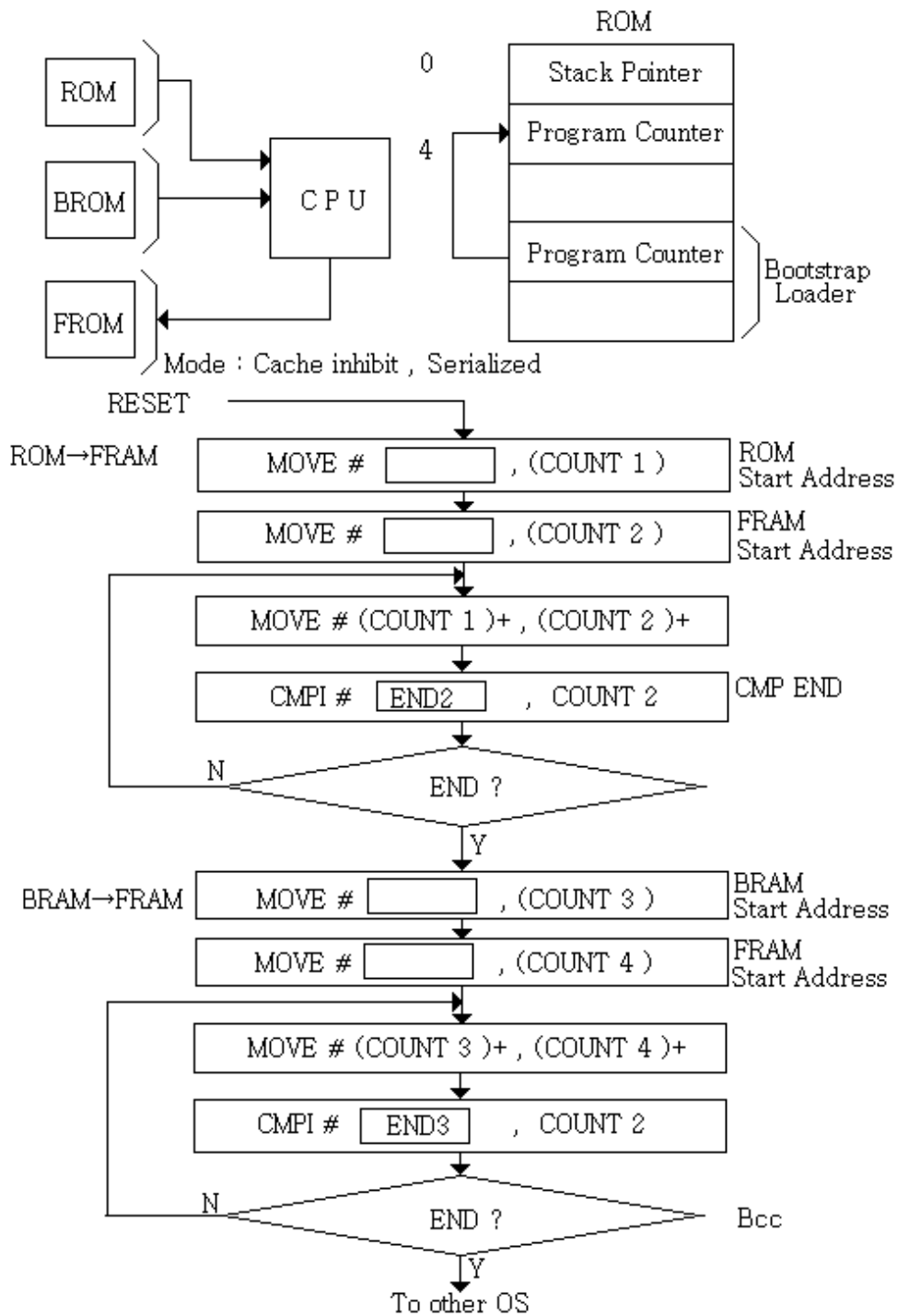


그림 3.11 부트스트랩 흐름도

Fig. 3.11 Flow chart of bootstrap

나. 타이머 초기화 프로그램

주기적인 인터럽트를 이용할 수 있도록 프로그래머블 타이머의 독립된 3개의 카운터 중 2개의 카운터를 Rate generator로 설정하였고, 입력 클럭을 분주하여 100[Hz]의 Rate generator 출력이 발생하도록 하였다.

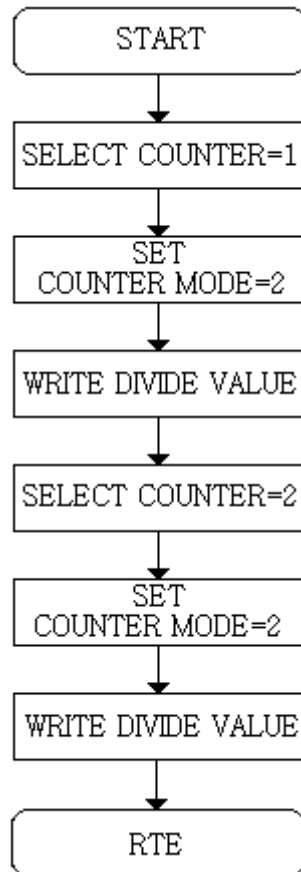


그림 3.12 타이머 초기화 흐름도

Fig. 3.12 Flow chart of timer initializing

다. 리얼 타임 클럭 초기화 프로그램

시스템 내에서 실시간 클럭을 이용할 수 있도록 RTC내의 클럭 레지스터 값을 초기화하여 RTC “on”후 실시간을 설정하도록 하였다.

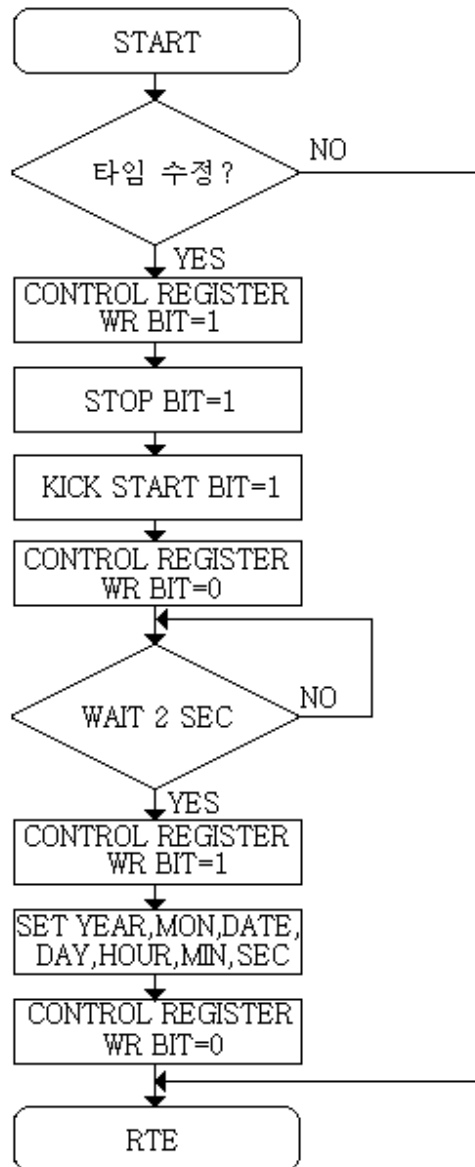


그림 3.13 리얼타임 초기화 흐름도

Fig. 3.13 Flow chart of real time clock initializing

라. 시리얼 커뮤니케이션 컨트롤러 초기화 프로그램

SCC 컨트롤러는 S/W적으로 H/W 리셋과 채널을 리셋하고 커뮤니케이션 프로토콜에 사용되는 SDLC 플래그, 어드레스 필드 설정 및 보레이트를 설정하여 동기식 RS422 또는 비동기식 RS232로 사용할 수 있게 되었다.

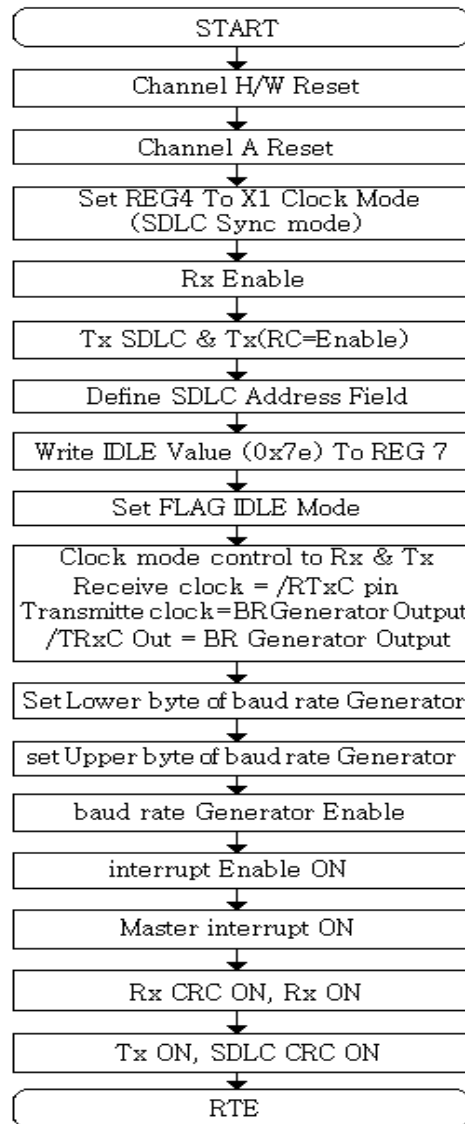


그림 3.14 시리얼 커뮤니케이션 컨트롤러 초기화 흐름도

Fig. 3.14 Flow chart of serial communication controller initializing

3.3.3. 인터럽트 프로그램

가. POWER DOWN 서비스 루틴(인터럽트 우선 순위 #7)

Power fail에 의한 Non - masking 인터럽트 루틴으로 주 메모리에 저장되어 있는 운전 중 데이터 및 시스템 상태를 백업용 SRAM에 저장하고 체크섬 값을 계산하여 저장한다.

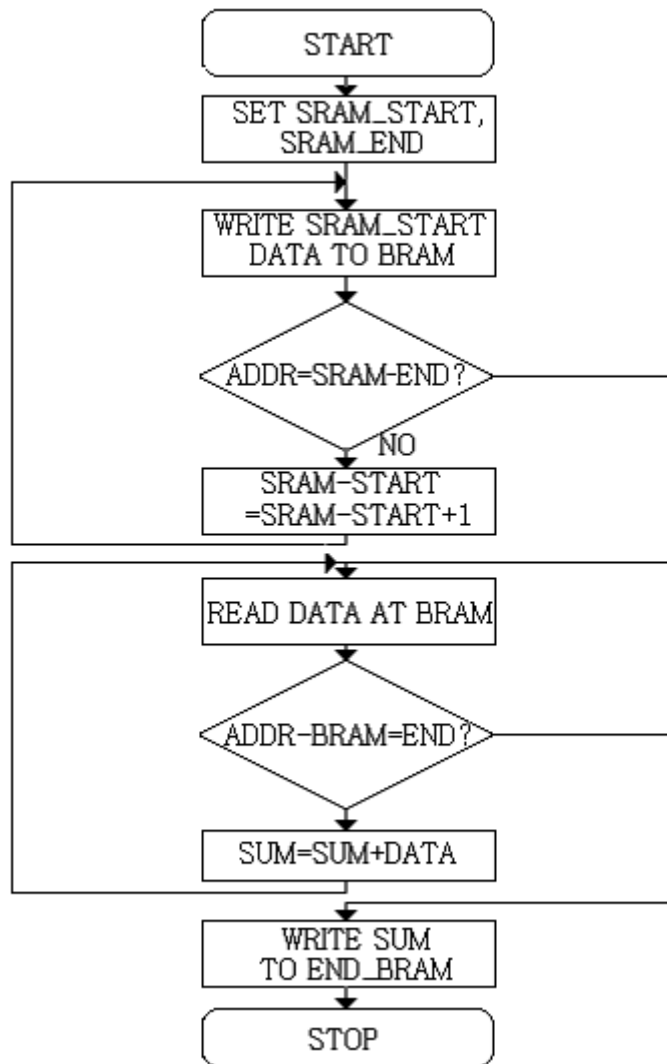


그림 3.15 파워 다운 흐름도

Fig. 3.15 Flow chart of power down

나. 스피드 센싱 서비스 루틴(인터럽트 우선 순위 #6)

Digital speed sensor 모듈에 의해 발생된 서비스 루틴으로 인터럽트 우선 순위 6번 이하의 인터럽트를 차단하고 24비트 카운터의 데이터 값을 리드한다.

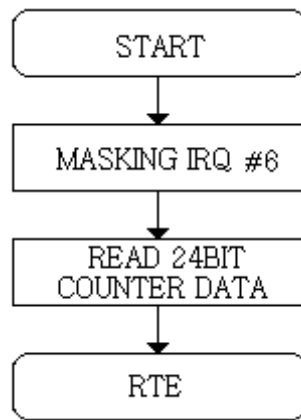


그림 3.16 스피드센싱 흐름도

Fig. 3.16 Flow chart of speed sensing

다. 타임 SCHEDULE 서비스 루틴(인터럽트 우선 순위 #5)

CPU내의 타이머에 의해 100[Hz]마다 주기적으로 발생되면 프로그램 그룹 Rate time에 따라 해당 task를 실행하도록 하였다.

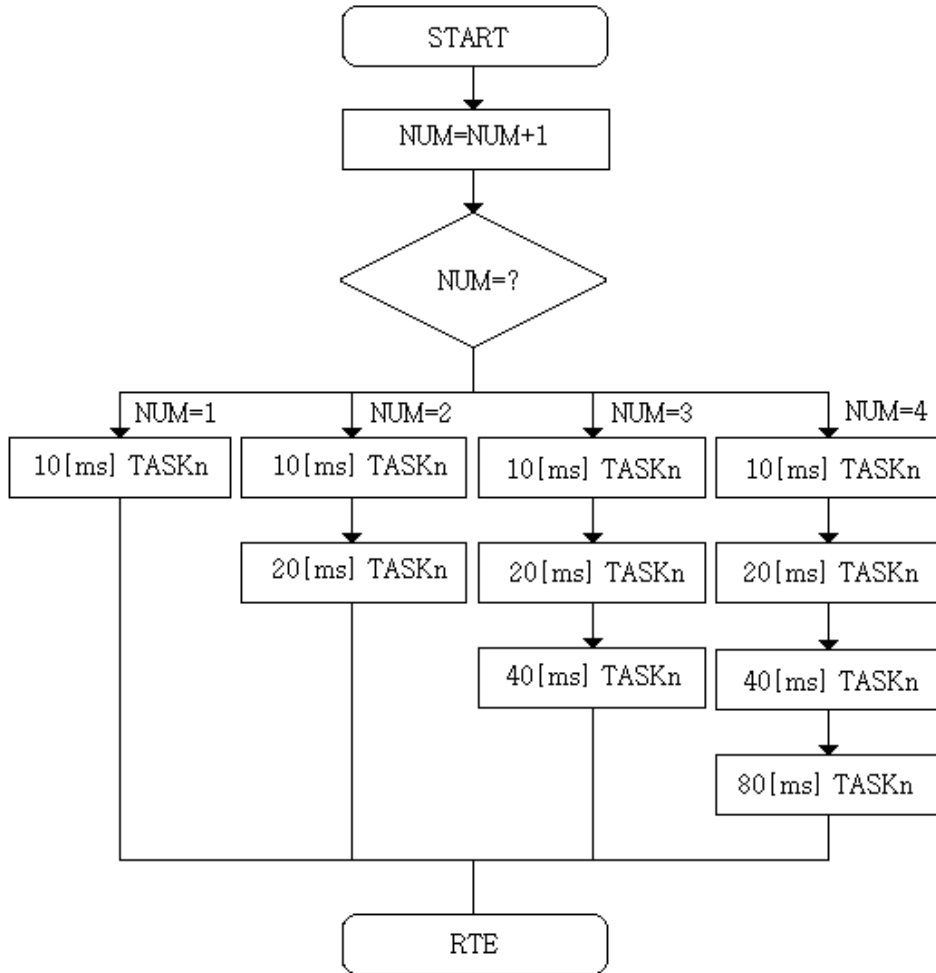


그림 3.17 타임스케줄 흐름도

Fig. 3.17 Flow chart of time schedule

라. SCC 리시버 서비스 루틴 (인터럽트 우선 순위 #4)

Inter communication에 사용되는 동기식 RS422이나 SIO 모듈에 사용되는 비동기식 RS232의 데이터를 수신시 인터럽트가 발생되며, 두 SCC 컨트롤러의 인터럽트 상태 플래그를 체크하여 Wired-or된 두 인터럽트를 수행 할 수 있다.

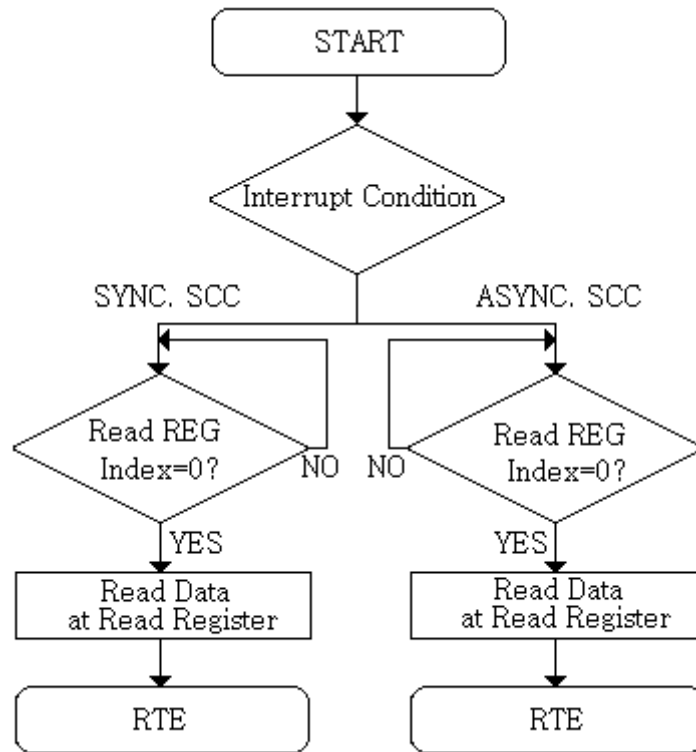


그림 3.18 SCC 리시버 흐름도

Fig. 3.18 Flow chart of SCC receiver

3.3.4 입/출력 제어 프로그램

가. 키패드 리드 프로그램

SIO 모듈의 24비트 키 값을 리드하여 각 키에 할당된 기능을 수행하도록 하였다.

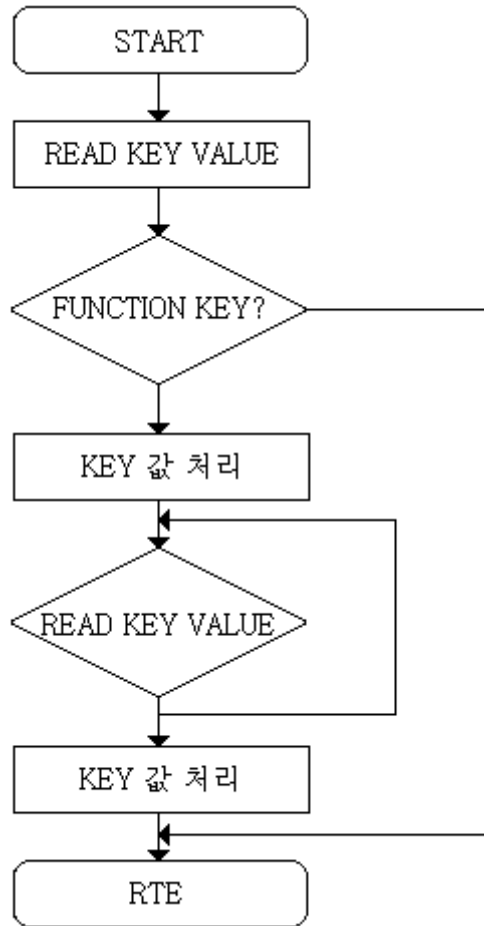


그림 3.19 키패드 흐름도

Fig. 3.19 Flow chart of keypad

나. RS232 프로그램

데이터 전송은 전송하고자 하는 데이터의 주소를 입력받아 데이터를 RS232 포트에 전송한다.

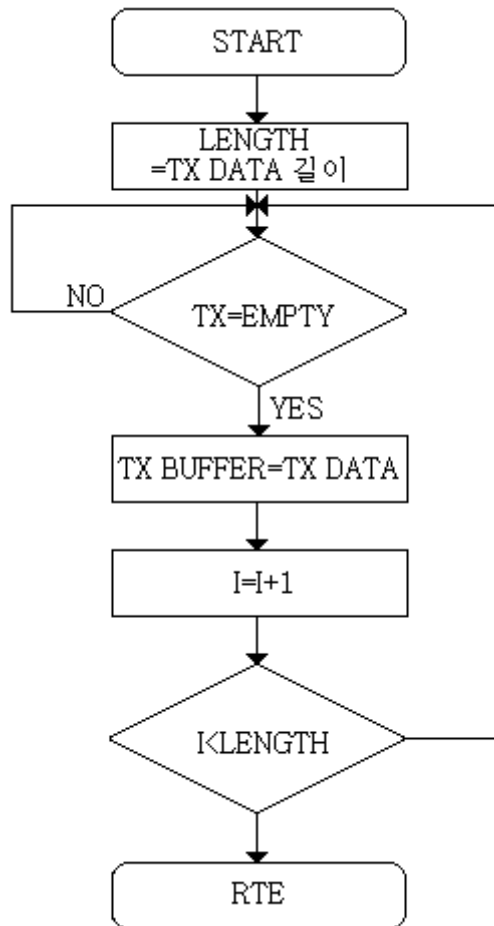


그림 3.20 RS232 흐름도

Fig. 3.20 Flow chart of RS232

다. LCD 컨트롤러 프로그램

LCD의 인터페이스 길이, 표시 행수, 문자 폰트를 설정 후 LCD 커서 형태를 결정하며, LCD 커서를 처음 위치로 이동 시킨 후 LCD에 기입된 모든 글자를 클리어 한다.

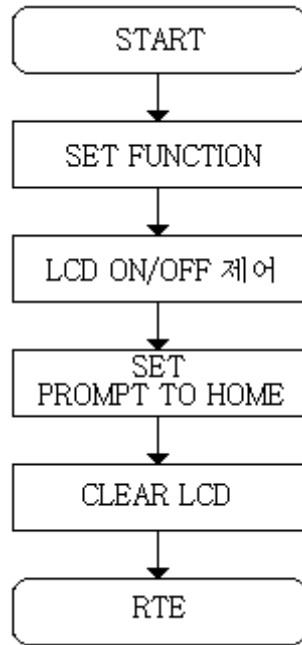


그림 3.21 LCD 컨트롤러 흐름도
 Fig. 3.21 Flow chart of LCD controller

라. 속도 변환 프로그램

Digital speed sensor 모듈로부터 입력받은 24비트 카운터 값을 RPM 계산식에 대입하여 속도 변화하는 프로그램이다.

$$RPM = (Fref \times Const) \div (Count \times Teeth)[rpm] \quad (3.1)$$

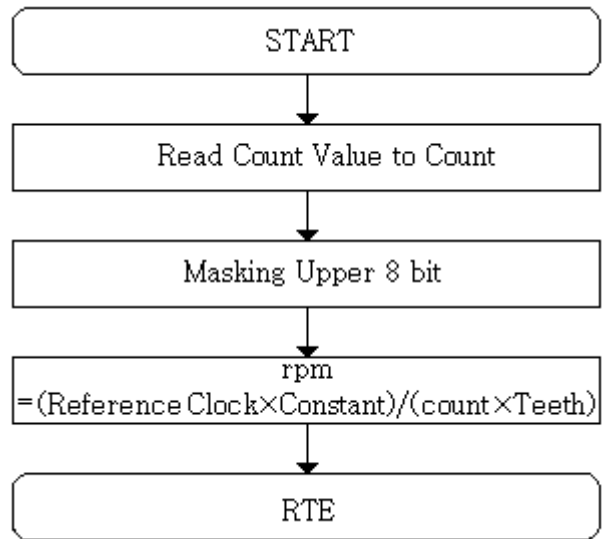


그림 3.22 속도 변환 흐름도

Fig. 3.22 Flow chart of speed regulation

3.3.5 인터 커뮤니케이션과 보팅 프로그램

가. 인터커뮤니케이션 프로그램

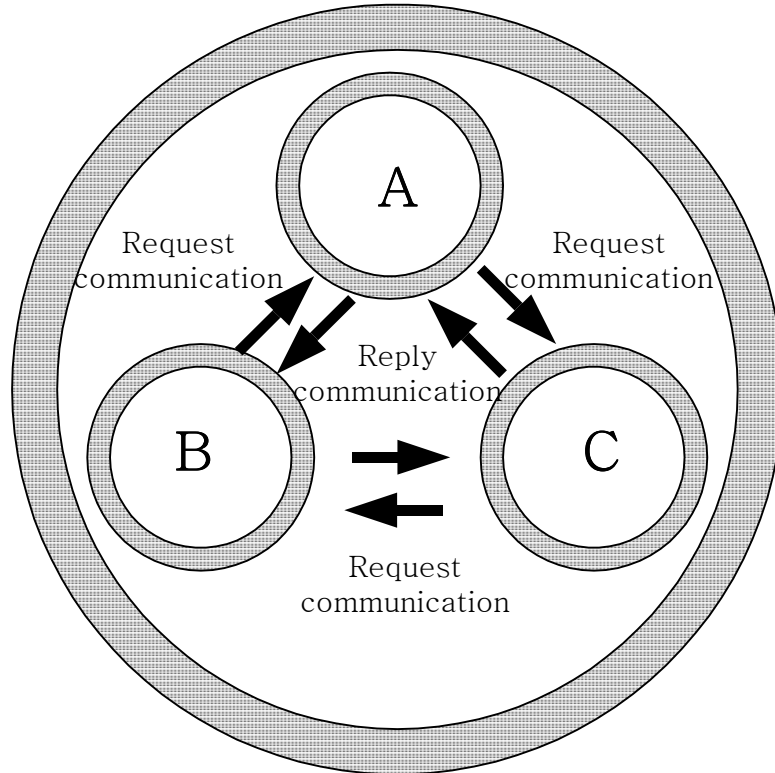


그림 3.23 인터 커뮤니케이션과 보팅 개념도

Fig. 3.23 Conception of inter communication and voting

Inter communication에 사용되는 프로토콜은 다음과 같이 이루어졌다.

SDLC FLAG	ADDRESS	COMMAND	CONTROL	TX LENGTH	nDATA BUFFER	SDLC FLAG
--------------	---------	---------	---------	--------------	-----------------	--------------

SDLC FLAG은 HEX 값으로 7Eh를 갖고, MARK IDLE과 FLAG IDLE로 사용 할 수 있으며, 현재의 시스템에는 FLAG IDLE 모드로 적용하였다.

어드레스 필드는 각 모듈에 할당된 어드레스로 각 CPU 모듈에 적용되는 어드레스는 CPU A(21H), CPU B(22H), CPU C(23H) 이다.

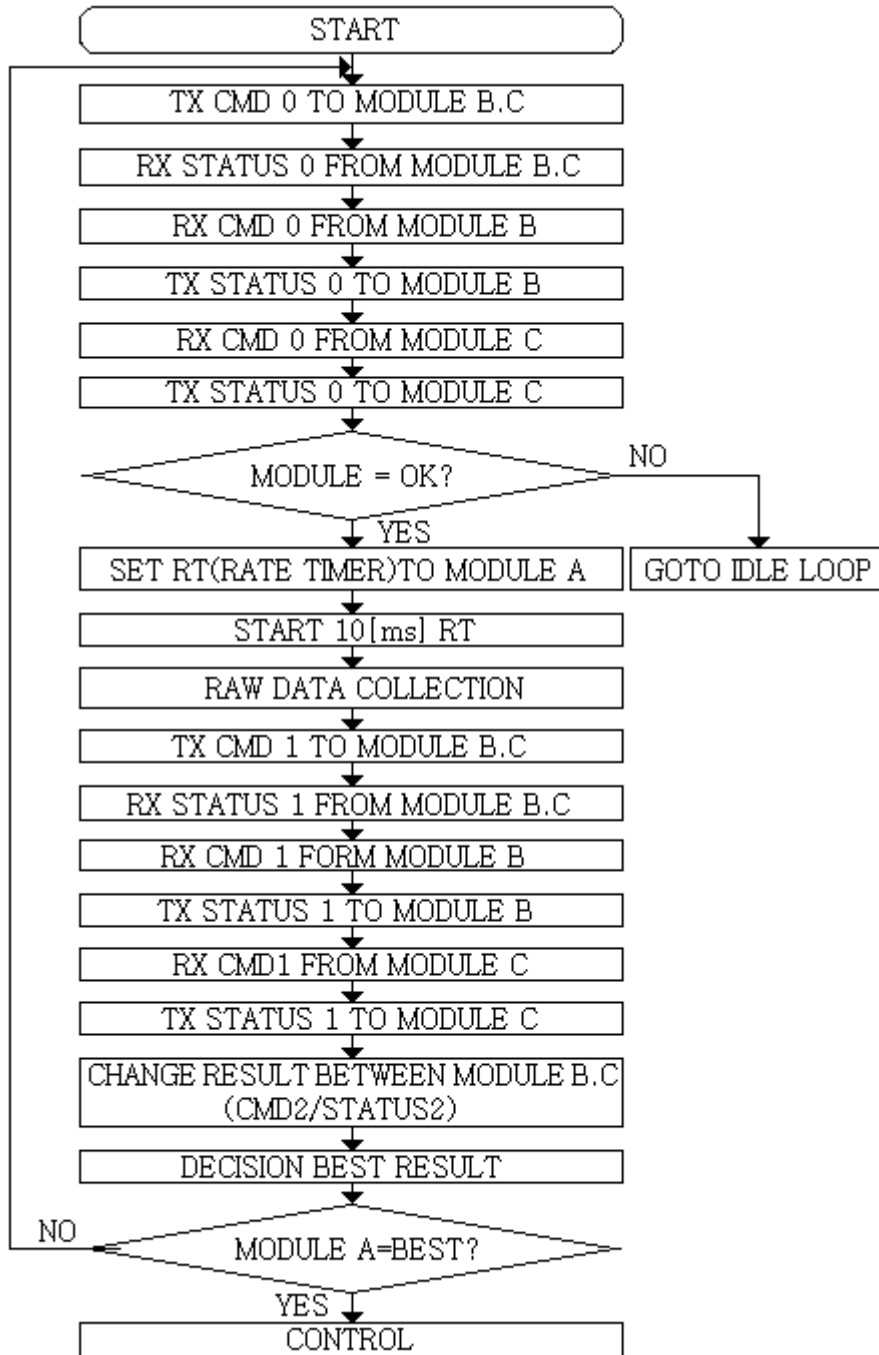


그림 3.24 인터 커뮤니케이션 흐름도

Fig. 3.24 Flow chart of inter communication

나. 보팅 프로그램

2 out of 3 voting은 3개의 CPU가 특정한 마스터 없이 스스로 데이터를 결정할 수 있다. 각 CPU에서 처리된 결과 데이터는 2 out of 3 voting을 사용하여 비교할 수 있으며, CPU가 정상적일 때 각 CPU에서 독립적으로 연산된 데이터는 항상 동일하다. 만약 한 CPU가 불량 데이터 일 때 다른 두 CPU에 해당하는 불량 데이터를 제거할 수 있으므로 항상 GOOD 데이터가 출력된다.

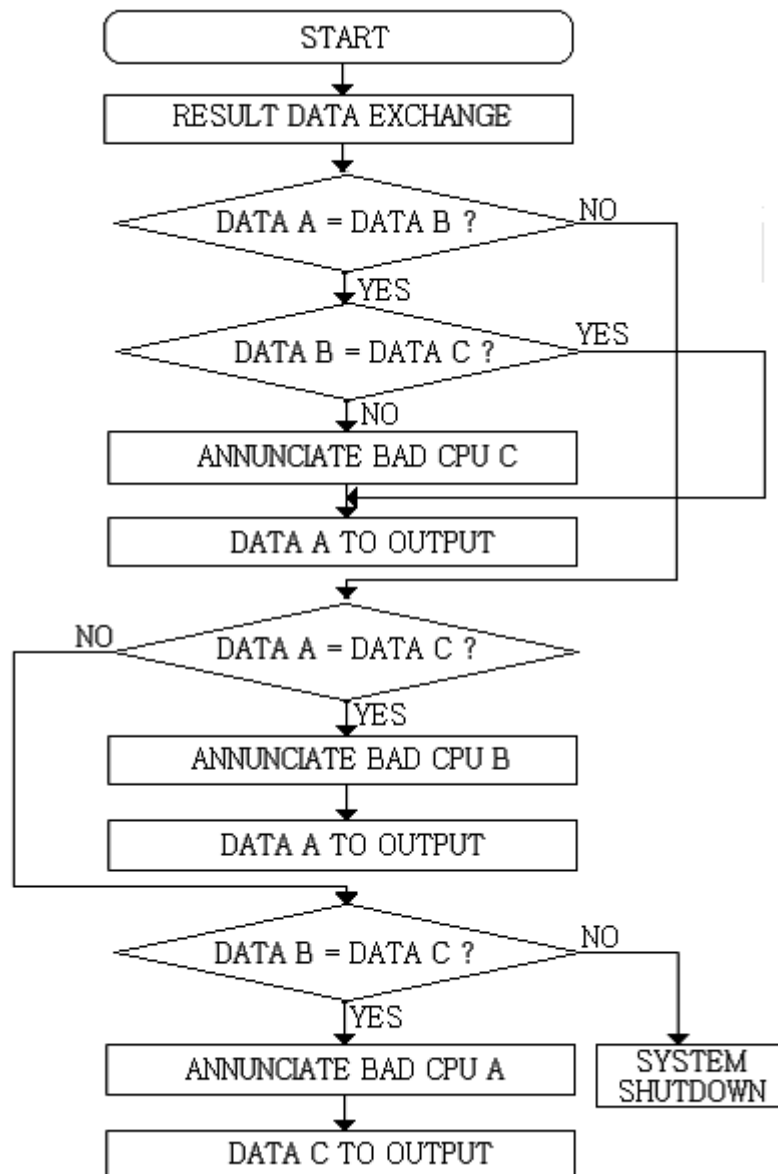


그림 3.25 보팅 흐름도

Fig. 3.25 Flow chart of voting

제 4 장 시스템 알고리즘의 구현

현재까지 개발된 디지털 조속기는 재래식 기계 유압식 조속기가 담당하던 조속 기능을 뛰어넘어 부하제어, 시동/정지 제어, 비상정지, 운전 파라미터의 화면 감시, 경보 이외에 다수의 보조기능들을 감당하고 있다. 이러한 디지털 조속기를 개발하기 위해서는 우선 몇 개의 기능으로 분류하고 다음 이를 하드웨어/소프트웨어적으로 구현하는 시스템 알고리즘을 개발하여야 한다. 만족스럽고 완벽한 시스템 알고리즘을 개발하기 위해서는 제어 및 하드웨어/소프트웨어에 관한 충분한 지식과 현장의 운전경험 등이 유기적으로 연결되어야 한다. 제한된 인력과 한정된 기간 내에 독자적으로 디지털 조속기를 개발 한다는 것은 대단히 어렵다. 따라서 본 연구에서는 Woodward Governor사의 디지털 조속기용 시스템 알고리즘을 분석하고 이를 수정 보완하여 복제주화력발전소용 디지털 조속기가 가져야 할 기능들을 열거하고 시스템 알고리즘을 개발한다.

전체 시스템은 10개의 복합 블록과, 이를 보조하기 위한 19개의 단일 블록으로 구성된다. 기능블록으로도 불리는 이 블록들의 결합은 디지털 조속기 시스템의 전체의 흐름을 가늠할 수 있게 한다. 각 블록의 기능은 소프트웨어적으로 구현되고 결합되어 실제적인 조속기의 기능을 수행하게 된다. 이러한 구성요소들을 빠짐없이 열거하고 시스템을 구성한 후 오류없이 시스템이 이루어졌는가에 대한 검증작업 또한 이루어져야 할 것이다. 여기에 열거된 기능들은 조속기로서 가져야 할 기본적인 기능이외에도 프로그램을 통한 기능의 추가 및 삭제도 가능하고, 입출력 지점의 디버깅 및 확장성도 가질 수 있도록 구성되어 있다.

4.1 복합 블록

여기서는 디지털 조속기 시스템의 전체의 흐름을 가늠할 수 있게 하는

복합 기능블록(Functional Block)의 신호 흐름과 그 블록이 가지는 기능을 설명한다. 입력 신호들은 속도신호, 기동/정지, 중·감속의 명령을 위한 점점입력과 발전기의 유효전력 측정 입력, 증기 제어 밸브의 변위측정입력 등 아날로그 신호입력을 받을 수 있다. 출력 신호들은 액추에이터로의 제어전류출력, 점점출력과 계기 장치를 구동시키는 아날로그 출력 등이 있다[24].

4.1.1 MPU_BLOCK

MPU_BLOCK 은 속도 감지를 위한 MPU로부터 속도신호를 입력받아 처리하는 블록이다. 이 블록은 I/O card로부터 디지털 값으로 속도신호를 입력받아 AN_IN_MPU 블록에서 이 신호 값을 처리하게 된다.

먼저 속도신호가 설정값 LOW_POINT나 HIGH_POINT의 값을 넘게 되면 SPD_SENSOR_BLOCK으로 속도신호의 오류를 알린다.

이는 ALARM_BLOCK과도 연결되어 운전자로 하여금 센서의 오류를 알게 한다. 오류를 발생시킨 센서의 복구 후 reset 신호를 받게 되고, 터빈의 기동 시에 속도센서가 터빈의 정지가 shutdown으로 인식되지 않도록 override 신호도 SHUTDOWN_BLOCK으로부터 받게 된다. "A" I/O에서 2개의 MPU가 사용되고, "C" I/O에서 2개의 MPU가 사용되게 된다.

4개의 신호가 모두 정상적인 값을 가질 때 SPD_SENSOR_BLOCK의 PASS_1, 2, 3와 4의 블록으로 보내어지게 된다. MPU의 속도 설정값의 LOW_POINT는 현장의 적용 상황에 따라 변동될 수 있으므로 TUNABLE_VALUE(*)로 되어 있다. 그림 4.1은 MPU_BLOCK의 기능블록을 도시한 것이다.

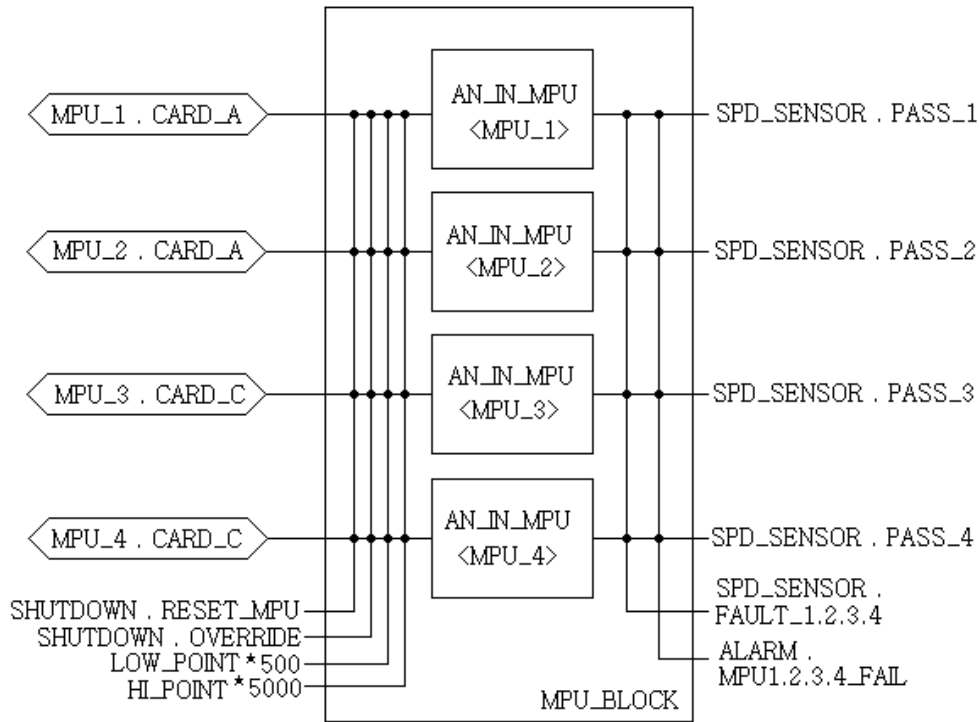


그림.4.1 MPU 블록

Fig. 4.1 MPU_BLOCK

4.1.2 SPD_SENSOR_BLOCK

SPD_SENSOR_BLOCK에서는 MPU_BLOCK에서 나온 속도신호를 받아 가장 높은 신호를 선택하도록 HSS_BUS_BLOCK으로 보내는 역할을 하는 블록이다. 이 과정에서 MPU/4~20[mA] board 오류의 발생이나, MPU_BLOCK에서 MPU의 오류 발생 신호가 들어오게 되면 AN_SW_BLOCK에서 전기스위치와 같은 동작을 행하게 된다. 즉, board의 오류나 MPU의 오류는 HSS_BUS로 가는 도중 입력에서 제거되게 된다.

이 과정에서 오류 경보음이 발생하고, 만약 4개의 board의 오류, 또는 4개의 MPU의 오류는 속도신호를 받아들일 수 없으므로 터빈을 정지하기

위해 SHUTDOWN_BLOCK의 PROBE_FL블록으로 shutdown 신호를 보내게 된다. 정상적인 값을 가지고 AN_SW블록을 통과한 속도 신호 값들은 HSS_BUS를 통과하게 되고, 터빈의 안정적인 운전을 위하여 가장 높은 속도 값 1개만이 제어의 파라미터로서 사용되게 된다. 이 HSS_BUS를 통과한 속도신호는 제어를 위한 여러 블록으로 보내어지게 되고, SPD_SENSOR_BLOCK에서는 마지막으로 위험회전수를 판단하는 기능을 수행하게 된다. HSS_BUS의 출력인 SPD_SENSOR의 BEST_SIG 신호는 2개의 위험회전수를 판단하는 비교블록을 통과하게 되고, 하나는 1차 지연요소를 가진 블록을 통과 하면서 속도신호가 잡음인지를 판단하게 된다. HSS_BLOCK을 통과한 신호가 잡음이 아니고, 위험회전수에 머물고 있을 경우, shutdown이 발생하도록 SHUTDOWN_BLOCK으로 shutdown신호를 보내게 된다. 일단 HSS_BLOCK에서 선택된 1개의 속도신호만이 다른 여러 블록으로 보내어지게 되는데, 아래와 같다.

ACK_BLOCK / ANAL_OUT_BLOCK / SERVICE_BLOCK
 SHUTDOWN_BLOCK / SPD_CNTRL_BLOCK / SPD_REF_BLOCK
 VALV_DRVR_BLOCK

그림 4.2에 SPD_SENSOR_BLOCK의 기능블록을 도시하였다.

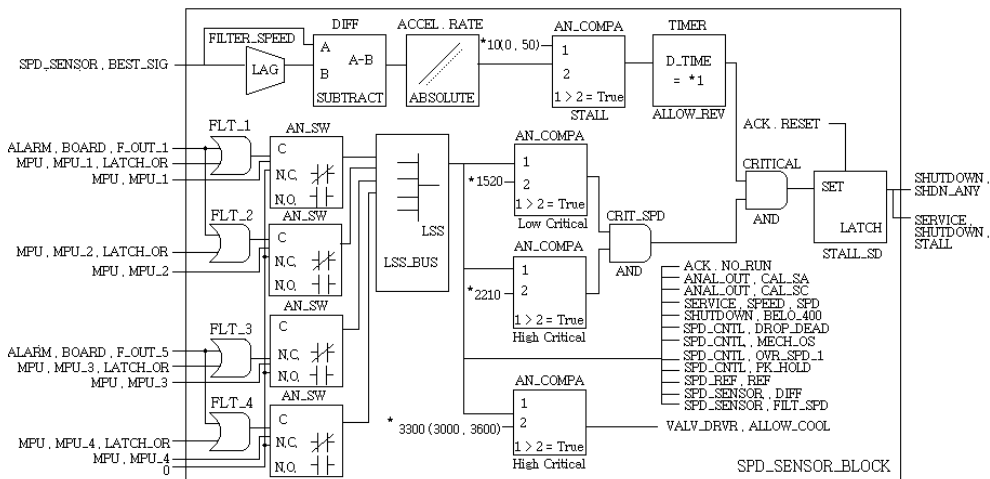


그림 4.2 스피드센서 블록

Fig. 4.2 SPD_SENSOR_BLOCK

4.1.3 ACKNOWLEDGE_BLOCK

ACKNOWLEDGE_BLOCK은 Reset switch 기능을 수행하는 블록이다. 어떤 이유로 터빈에 shutdown이 발생하게 되면 ACT_OUT_BLOCK에서 액츄에이터를 shutdown 상태로 유지하게 된다. 이때에 ACK신호가 가지 않으면 액츄에이터는 터빈을 계속 shutdown된 상태로 지속하게 된다. 또한, 각 기능블록의 reset 기능도 수행하여 각 기능블록내의 latch들은 reset시키는 역할을 하게 된다. 이 블록의 입력은 조속기 Control rack panel의 Reset button을 누르거나, 외부 점점입력 또는 서비스 모드에서 reset 시킬 수 있다. 이러한 입력들이 들어가게 되면 내부의 reset기능은 One shot와도 같은 기능을 하는 블록을 통과하여 ACT_OUT_BLOCK / SERVICE_BLOCK과 모든 블록으로 신호가 보내진다. 그림 4.3은 ACKNOWLEDGE_BLOCK의 기능 블록선도이다.

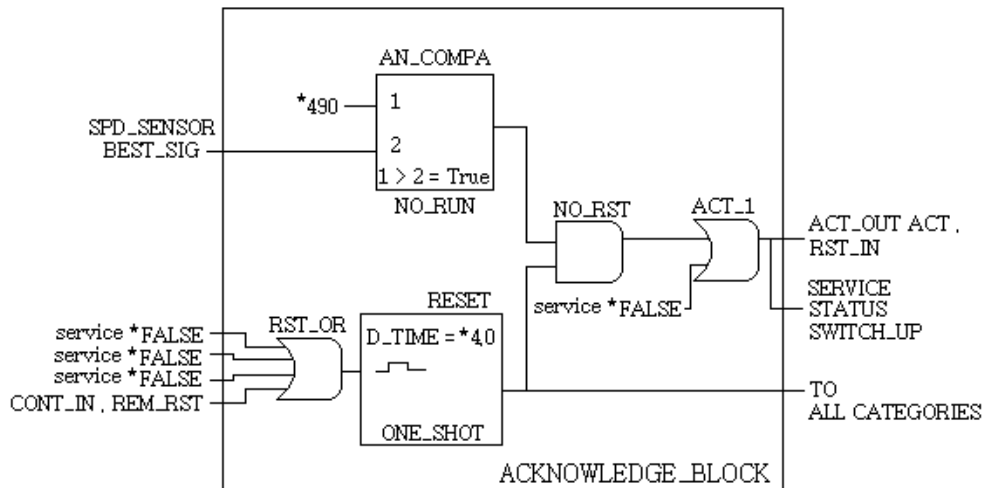


그림 4.3 ACKNOWLEDGE_블록

Fig. 4.3 ACKNOWLEDGE_BLOCK

4.1.4 VALV_DRVR_BLOCK

VALV_DRVR_BLOCK에서 크게 두 가지의 기능블록이 중요한 역할을 담당한다. 하나는 ramp블록이고, 하나는 dither블록이다. 기능블록의 자세한 기능은 블록선도에서 설명되고 있다. 이 블록에서 ramp의 기능은 터빈의 제어밸브의 일정 지점까지의 변화비율을 나타내고 있다. 각 rate의 비율에 따라 밸브의 움직이는 증가율이 바뀌게 되고 이를 역으로 하면 감소율이 된다. 또한, dither블록에서는 서보밸브의 고착을 방지하기 위하여 일정 비율의 진동을 주게 된다. 이 블록에서는 LSS_BUS가 채택되어 있다. 이는 입력값의 제일 작은 값이 선택되어 출력되게 된다. ramp 블록에서부터의 출력은 LSS_BUS를 통과하게 되고, 이 출력은 dither블록으로 들어가서 이 신호는 액츄에이터를 구동시키는 블록인 ACT_OUT_BLOCK으로 신호를 보내게 된다. LSS_BUS를 통과하는 신호들 중 Allow_Cool 신호는 세 가지 조건이 만족되면 LO_LIM신호를 입력되어 변수 설정값이 반영되게 한다. 그림 4.4는 VALV_DRVR_BLOCK의 기능블록선도이다.

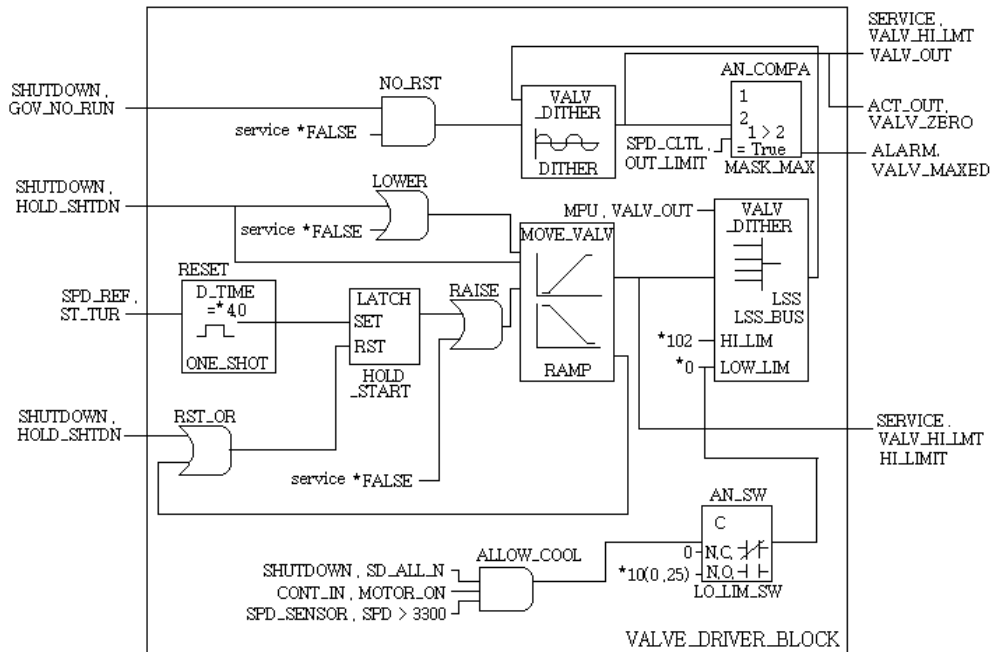


그림 4.4 밸브 드라이버 블록

Fig. 4.4 VAVL_DRVR_BLOCK

4.1.5 ACT_OUT_BLOCK

이 기능블록은 ACT_ZERO / ACT_SPAN의 블록과 ACT REDUNDANT MANAGEMENT의 세 블록으로 나뉘어져있다. ADD블록과 MULTIPLY블록을 통과하면서 개인과 옵션을 가진 값을 가지고 액추에이터로 들어가게 된다. ADD블록의 입력은 동조 가능한 변수입력과, VALV DRVR BLOCK의 dither 블록으로부터 입력을 받게 된다.

ACTUATOR_REDUNDANT_MANAGEMENT블록에서의 신호는 “A”와 “C” I/O의 Final driver card로 출력을 내보내게 된다. “A” I/O의 Final driver card에는 Redundant management를 위해 Coil_1과 Coil_2 두 개의 코일을 구동할 수 있도록 설계되어 있다. “C” I/O도 마찬가지로 두개

의 코일을 구동할 수 있도록 설계되어 있다.

현재 운전 중인 코일의 상태로 서비스 모드에서 나타내도록 하였고, “A” card의 fail은 “C” card가 담당하고, Coil_1과 Coil_2로 backup이 되어 있어 신뢰성을 증가시킨다. 그림 4.5은 ACT_OUT_BLOCK의 기능 블록선도를 보여준다.

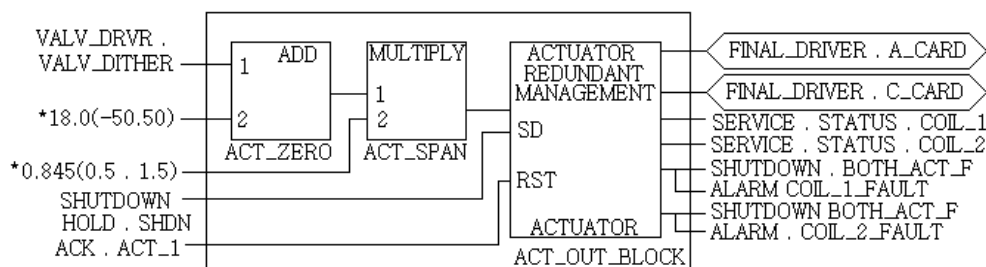


그림 4.5 액추에이터 출력 블록

Fig. 4.5 ACT_OUT_BLOCK

4.1.6 SHUTDOWN_BLOCK

SHUTDOWN_BLOCK은 터빈이 shutdown될 조건을 만족하게 되면 터빈을 정지시키는 기능을 가지는 블록이다. CONT_IN블록에서 Trip신호가 들어오게 되면 수동 trip이 latch되고, 이는 두 개의 전원공급이 상실되거나, 두 개의 ACT. 코일이 상실되거나, overspeed 1, 2와 위험 회전수내에서의 지속적인 운전이 감지되었을 경우 터빈의 shutdown을 발생시키게 된다. 그림 4.6은 SHUTDOWN_BLOCK의 블록선도이다. 전원공급이나 ACT 코일의 상실은 두 개가 모두 상실될 경우에는 shutdown을 발생시키게 된다. 이들은 일단 latch블록에 set되어 reset입력이 들어올 때까지 shutdown입력신호를 홀드하게 된다. shutdown신호는 다음의 기능블록들로 신호를 보내서 터빈을 정지 시키게 된다.

KW_CNTL_BLOCK / ACT_OUT_BLOCK / AUTO_SPD_BLOCK
 SPD_REF_BLOCK / VAVL_DRVR_BLOCK / CONT_OUT_BLOCK
 MPU_BLOCK

이 SHUTDOWN_BLOCK은 MPU를 reset시키기 위하여 MPU의 속도가 어느 일정 속도 이하로 내려간 후나, ACK블록의 reset입력을 받아 네개의 MPU를 Rest 시키는 기능과, shutdown후에 터빈을 다시 기동시킬 경우 낮은 속도의 감지를 override시키는 기능을 동시에 가지게 된다.

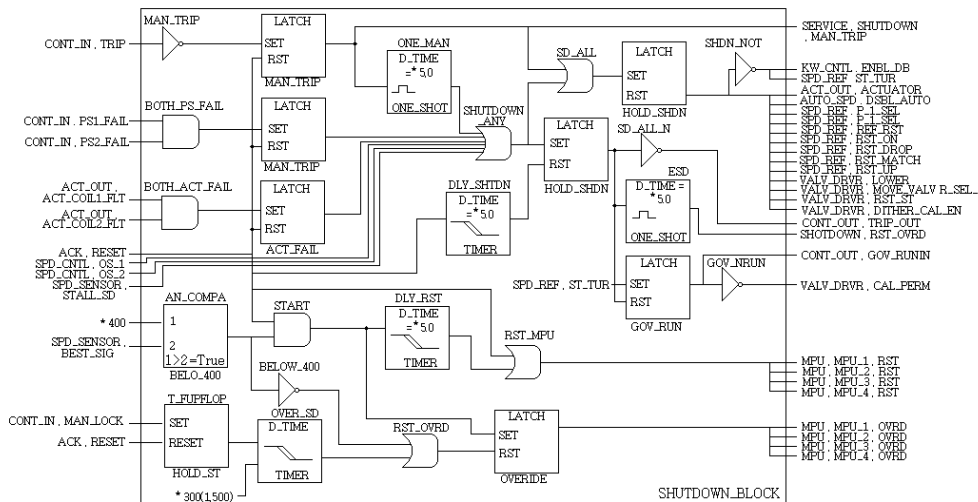


그림 4.6 첫 다운 블록
 Fig. 4.6 SHUTDOWN_BLOCK

4.1.7 ANALOG_IN_BLOCK

ANALOG_IN_BLOCK은 4~20[mA]의 전류신호를 내는 현장의 센서들로부터 입력을 받는 블록이다. 전류신호들은 디지털 신호로 바뀌어 지게 되고 각각 "A" I/O와 "C" I/O로 들어가게 된다. 그림 4.7은 ANALOG_

IN_BLOCK "A" Kernel의 블록선도이고, 그림 4.8은 ANALOG_IN_BLOCK C Kernel의 기능 블록선도이다. 입력신호는 ANAL_IN블록을 통과하게 되는데, 여기에는 기기로부터의 과대, 또는 과소한 입력을 제거하기 위해 이용되고, 각각 상한값과 하한값을 설정할 수 있고 변경도 가능하다. 만약 설정값 이상의 값들이 입력되게 되면 경보음을 발생시키고, 운전자가 ACK블록에서 reset을 시켜야 제 기능으로 복귀된다. 이 블록을 통과한 입력 값은 ADD블록과 MULTIPLY블록을 통과하며 게인값과 오프셋값을 조정하여 ANALOG입력을 필요로 하는 아래와 같은 블록으로 보내어준다.

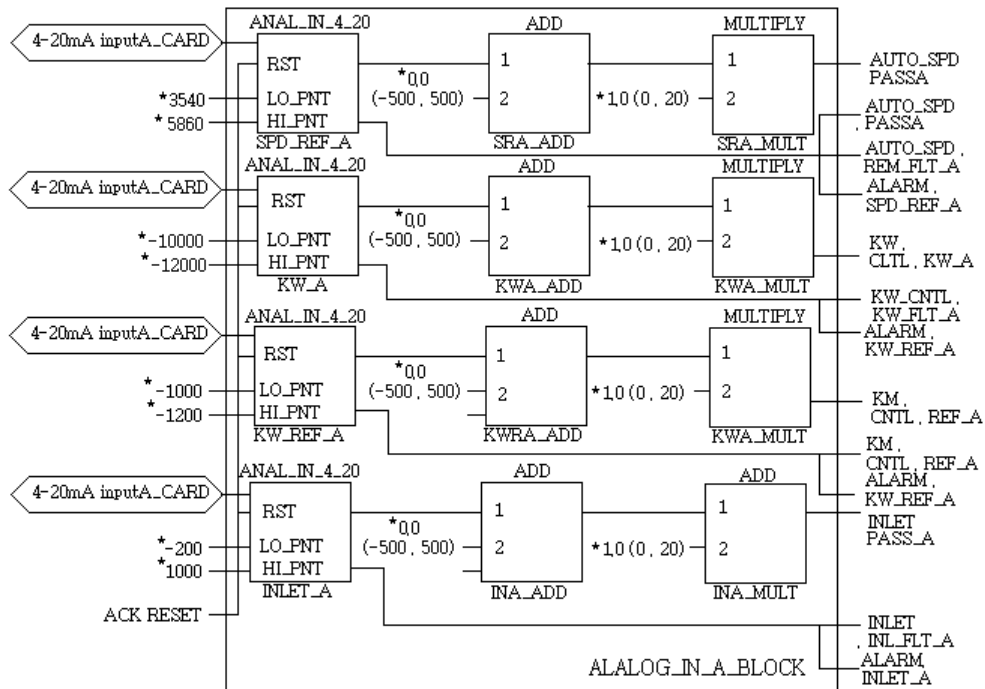


그림 4.7 아날로그 입력 A 블록

Fig. 4.7 ANALOG_IN_A_BLOCK

AUTO_SPD_BLOCK / KW_CNTL_BLOCK / INLET_PRESS_BLOCK
위와 같은 기능블록들로 4~20[mA]의 입력신호를 보낼 수 있으나, 현장

의 환경에 따라 변경 및 재 프로그래밍이 가능하고, 계인 및 옵션값들도 수정이 가능하게 되어 있다.

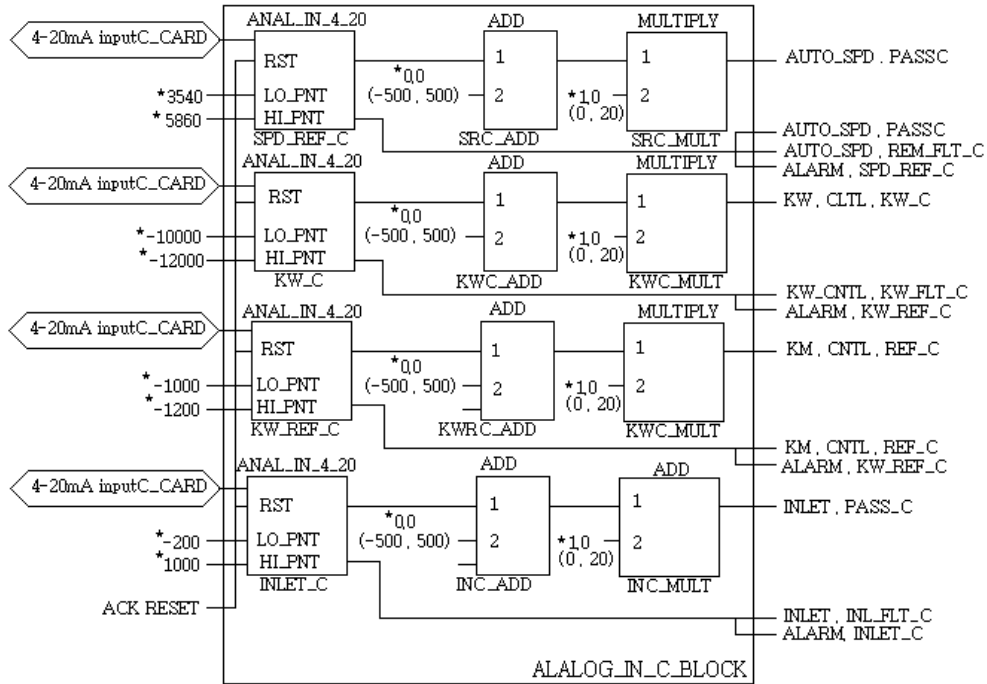


그림 4.8 아날로그 입력 C 블록
Fig. 4.8 ANALOG_IN_C_BLOCK

4.1.8 CONTACT_INPUT_BLOCK

이 블록은 외부의 점점입력-운전자가 버튼을 조작하거나 스위치를 조작하는 등의-조작을 통해서 터빈의 운전하는 기능을 가진 블록이다. 외부의 점점의 조작은 "A" I/O와 "C" I/O의 입력보드를 통해서 들어오고, 각각의 채널을 설정하여 그 점점의 기능이 그에 해당하는 기능블록으로 출력이 되게 된다. 이런 CONTACT_INPUT블록은 하이입력과 로입력으로 그 점점입력에 사용에 따라 분류를 해놓았다. LOW블록의 입력은 다음과 같다.

START / TRIP / RATED SPEED ENABLE / SPEED RAISE LOWER
 REMOTE CONTROL ENABLE / OVERSPEED-TEST ENABLE KW
 CONTROL ENABLE / POWER SUPPLY #2 STATUS GEN. BRK.
 STATUS POWER SUPPLY #1 STATUS
 HIGH블록의 입력은 다음과 같다.

REMOTE RESET / HYDRAULIC FILTER HIGH DP

그림 4.9는 CONTACT_INT_BLOCK_LOW의 기능 블록선도이고 그림 4.10은 CONTACT_INPUT_BLOCK HIGH의 블록선도이다. 여기서 Hydraulic filter high DP는 액츄에이터로 가는 유압 라인상의 필터의 차압을 감지하여 입력으로 받게 된다.

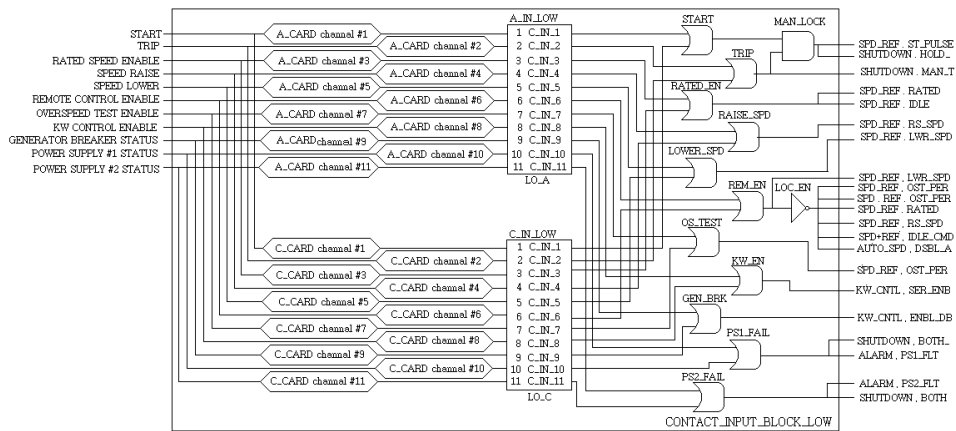


그림 4.9 콘택 입력 로 블록

Fig. 4.9 CONTACT_INPUT_LOW_BLOCK

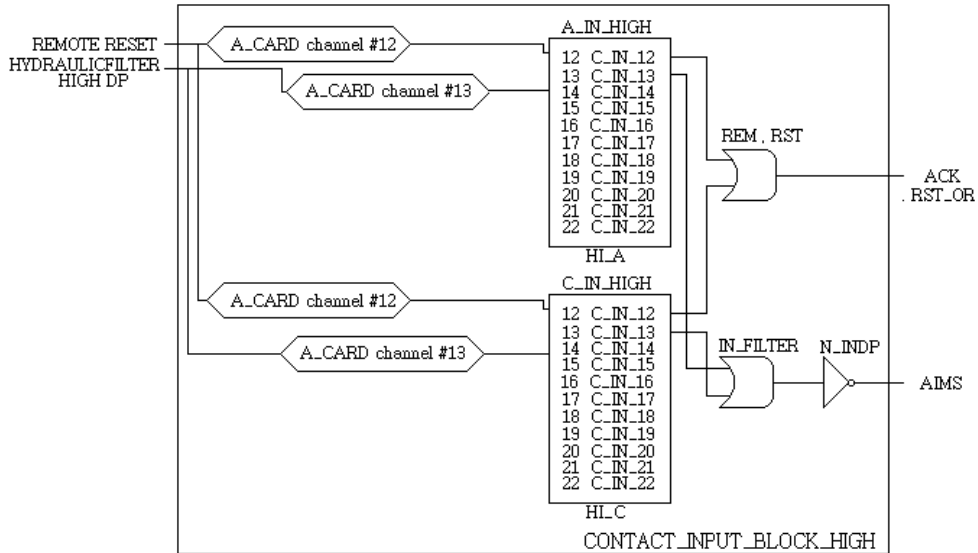


그림 4.10 컨택 입력 하이 블록

Fig. 4.10 CONTACT_INPUT_HIGH_BLOCK

4.1.9 CONT_OUT_BLOCK

이 블록은 외부의 접점출력을 위하여 그 기능이 사용되고 “A”와 “C”의 두개의 출력을 가지고, 정의된 접점출력의 채널을 통해 각각 출력된다. 그림 4.11는 CONT_OUT_BLOCK의 기능을 보여주고 있다. 이 기능블록은 출력의 화장을 위해 하드웨어가 더 많은 채널을 가질 수도 있고, 경우에 따라 재 프로그래밍도 가능하다. 여기서는 5가지의 접점출력을 정의하였고 추후 정정 및 디버깅도 가능하다.

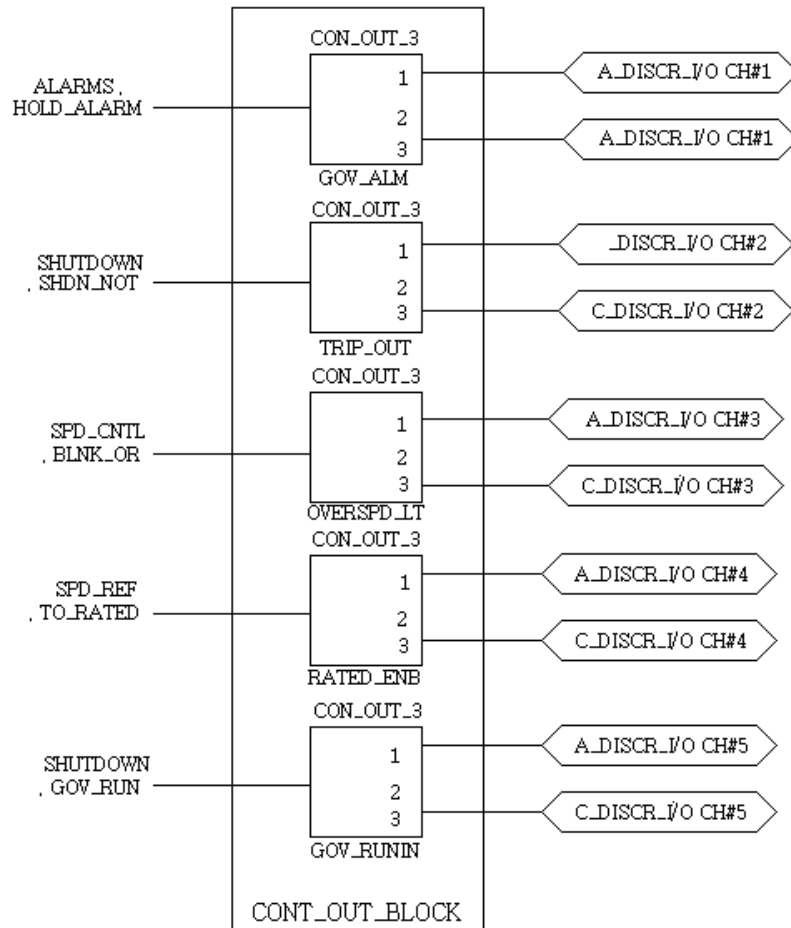


그림 4.11 컨택 출력 블록

Fig. 4.11 CONTACT_OUT_BLOCK

4.1.10 SERVICE_BLOCK

이 블록은 두 가지의 종류로 나눌 수가 있다. 하나는 Service alarm category이고 하나는 Monitor category이다. Monitor category에서는 디지털 조속기 내부의 프로그램 된 변수들의 값들의 변경 및 감시가 가능

하고 운전자의 CONT_IN블록에서의 입력이 아닌 service모드에서의 운전 모드의 변경 등이 가능하게 된다. 이 Monitor category의 사항들은 아래와 같다.

SHUTDOWN / RESET / SPEED / INLET PRESSURE / KILOWATT
HIGH VALVE LIMIT / STATUS

이런 기능들은 제어판넬의 키패드의 조작에 의하여 상·하, 좌·우로 옮겨 감으로써 원하는 항목으로 이동을 할 수가 있게 된다.

4.2 단일 블록

여기서는 디지털 조속기용 시스템 알고리즘 내에서 중요한 역할을 담당하는 단일 기능블록들의 역할 등을 자세하게 설명한다. 이 기능들의 대부분은 다소 전자소자와 같은 기능의 설명을 하지만, 그 자체가 가지는 의미는 이 블록에서 가지는 프로그램의 특성을 설명한 것일 뿐이다. 이하에 설명되는 기능들이 실제 CPU 모듈내의 메모리에 들어가게 되는 기능 프로그램으로서 외부로부터 입력되는 신호들을 어떤 방식으로 처리할 것인가를 나타내고 있다.

4.2.1 AI_MPU (ANALOG INPUT DIGITAL MPU CHANNEL)

이 블록은 Digital speed sensor H/W보드로부터 입력을 받는다. 이것은 두 개의 출력을 생성할 수 있는데, 하나는 속도에 비례하여 검출된 [Hz]나 [rpm]이고, 다른 하나는 속도 미분이다. 미분 출력은 초당 [Hz]나 [rpm]의 변화율이고, DERIV_EN의 부울대수 입력으로 선택되어질 수 있다. 만약 최고속도가 High set point (HI_SP)를 넘어가게 되면 High latch output(HI_LATCH)가 true로 가게 된다. High latch output은 입력이 HI_SP를 넘고 있는 동안은 RST의 입력에 관계없이 계속 true를 유

지한다. 만약 입력이 High set point 이하로 내려가게 되더라도 High latch output은 Reset input이 true 될 때까지 기다린다.

그 후에 High latch output도 false로 가게 된다. Low set point (LO_SP) 입력은 주파수 범위 입력이 false이면, Low latch output이 true가 된다. 만약 Low-limit override 입력이 true이면, Low latch output은 true가 될 수 없다.

Low-limit override 입력도 High latch output에 아무런 영향을 주지 않는다. Low latch output이 true로 가게 되면 입력이 Low set point로 올라가더라도 계속 true로 있게 된다. 만약 입력이 Low set point 보다 높게 되고 reset이 true이면 Low latch output도 false로 가게 된다. Low latch output이 false로 가서 계속 유지하려면 Low-limit override input을 true로 하고 reset 입력을 true로 해야 한다. 만약 Reset input이 true이면 High latch와 Low latch output로 Non latching된다. latch가 true로 가는 출력을 지연시킨다. 입력값은 Set point 를 LATCH_DLY 시간보다 더 길게 해야 한다. 만약, HI_LATCH와 LO_LATCH나 오류 output이 true이면 부울대수 출력이 true가 될 것이고, 그렇지 않으면 false이다.

board 오류 output은 부울대수 출력으로 true일 경우 프로세서에게 Digital speed sensor H/W board의 특정채널에 고장이 났다고 지시한다. 어떤 한 채널의 고장도 또한 오류 출력을 true로 가게 한다.

block의 최대 출력이 FREQ_RNGE 값의 최대값이 된다. 또한 최소값은 FREQ_RNGE의 최소값이 되고 그 이하로 내려갈 경우 MIN_SPEED 부울대수 출력이 true가 된다. monitor 출력은 콘트롤에는 사용되지 않고 속도 지시용으로만 사용된다. S_FILT와 D_FILT 입력은 Speed filter와 Derivative filter의 시정수라고 볼 수 있다. S_FILT와 D_FILT의 최소값은 aliasing과 noise 방지를 위해서 ALMPU MOE_BLOCK의 Rate group sample time보다는 반드시 커야한다. ACCEL LIM 입력은 정(+)이나 부(-)

의 속도 변화율을 속도 필터나 미분계산에 가기 전에 제한함으로써 노이즈를 방지한다.

4.2.2 LAG (LAG FILTER BLOCK)

이 블록은 LAG_TAU에서 입력되는 값에 따라서 지연시키는 블록이다. DERIV 출력 field는 입력의 변화율이다.

$$\text{OUTPUT} / \text{INPUT} = 1 / (1 + ST)$$

여기서, $T=1 / 2\pi f = \text{LAG_TAU}$ (1차 시간지연)

4.2.3 SUBTRACT

이 블록은 뺄셈항목으로서 피감수에서 감수를 빼게 된다.

$$\begin{aligned} \text{OUTPUT} &= \text{IN}_1 - \text{IN}_2 \\ &= \text{MINUEND INPUT} - \text{SUBTRAHEND INPUT} \end{aligned}$$

4.2.4 ABSOLUTE

이 블록은 입력과 출력의 절대값을 취하는 블록이다. 이 블록의 출력은 항상 입력의 부호에 관계없이 양이다.

4.2.5 HSS_BUS (HIGH SIGNAL SELECT BUS)

출력은 입력값에서 가장 높은 값과 같고, 들어온 최대값은 High limit (HI_LIM)보다 높지 않고, Low limit indicator 부울대수 출력(LO_LIM_I)이 true가 된다. 만약 들어온 입력의 최대값이 HI_LIM보다 높게 되면 HI_LIM이 출력을 결정하게 되고 HI_LIM_I 부울대수 출력은 true가 된다. 이 HSS의 각각의 입력에는 In control input이 있어 입력이 가장 높은

값을 선택하여 그에 해당하는 부울대수 값이 true가 된다.

4.2.6 AND

모든 입력이 true일 경우에만 출력이 true이다. 어느 하나의 입력이 false이면 출력은 false이다. 추가적인 입력으로 repeat 명령을 사용한다. (최대 600개)

4.2.7 OR

어떤 입력이 true일 경우 이 블록의 출력은 true가 된다. 모든 입력이 false일 경우 이 블록의 출력은 false가 된다. 추가적인 입력으로 repeat 명령을 사용한다. (최대 600개)

4.2.8 LATCH

이 블록은 부울대수 입력(trigger)을 보고 있다가 입력이 true일 때 출력 (latch)을 set한다. 만약 입력이 false이고 RST가 true이면 출력은 false가 될 것이고, 둘 다 true가 되면 출력은 true가 되어 입력은 RST에 대해 우선권을 가진다.

4.2.9 ACT_RM (Actuator Output Redundancy Management)

이 블록은 Actuator driver block으로 가고, 그 Driver block은 출력을 조정한다. Actuator driver block은 각각 저마다 Actuator driver 블록의 오류 출력에 연결되어 있다. 액츄에이터 출력(ACT_1_OUT, ACT_2_OUT)은 각각의 액츄에이터 driver의 입력으로 연결되어 있다. 액츄에이터 경고 출력

(ACT_1_ALM, ACT_2_ALM)은 일치하는 피드백 입력이 true로 가면 경보를 출력한다. 입력은 0~100[%] 사이의 값이다.

리셋(RST) 입력이 true이고, 일치하는 피드백 입력(FEEDBACK_1, FEEDBACK_2)이 false일 경우 경고 출력은 false가 된다. shutdown입력 (SD_1, SD_2)이 참이면 관련된 ACT_1_SD나 ACT_2_SD은 0으로 가게 된다. 이는 Intergrating actuator로는 작동되지 않는다.

4.2.10 ADD

이 블록은 입력을 서로 더하는 기능이다. 입력은 음의 값일 수도 있다.

$$\text{OUTPUT} = \text{IN}_1 + \text{IN}_2 \dots$$

추가적인 입력을 원하는 경우 repeat 명령을 사용한다. (50까지)

4.2.11 DITHER

이 블록은 Software dither와 calibration 두 가지 기능을 가진다. Software dither 기능은 아날로그 출력을 dither(INPUTDI_IN)의 입력 값에 따라 입력 신호의 상하로 진동시키기 위해 사용된다. 이런 상태는 permissive와 enable 입력(CAL_ENBL과 CAL_PERM)가 false일 때 가능하다. calibration을 할 때는 입력 신호가 출격을 드라이브하기 위해 수동위치로 되어있어야 한다. 이런 상태는 CAL_ENBL과 CAL_PERM이 모두 true일 경우에 가능하다.

Dither rate(D_RATE) 입력은 dither의 주파수를 설정한다. 알람 출력은 CAL_ENBL이 false일 경우 true로 가게 된다. ALM은 입력이 바뀔 때까지 true로 가고 RST 입력이 true로 갈 때까지 true를 유지한다.

4.2.12 RAMP

이 블록은 어떤 특정 값이 선택되어지면 그 지점까지 입력한 기울기를 가지고 증가하거나 감소한다. 이는 전원 투입 시 특정한 값으로 초기화될 수 있다. 출력은 특정 단위로 표시되고, 기울기는 단위/초로 나타낸다. P_SEL_X 입력은 엔코더와 같은 역할을 하고 USE_REV가 false일 때 P_SEL_1은 P_SEL_2에 대해 가장 높은 우선권을 가지고 가장 늦은 P_SEL_X는 가장 낮은 우선권을 가진다. ramp가 움직이는 비율은 DFLT_RATE, R_SEL_X와 R_SP_X에 의해 결정된다. 만약 R_SEL_X와 R_SP_X가 없으면 ramp는 DFLT_RATE에 의해 움직인다. 만약 R_SEL_X와 R_SEL_1이 가장 높은 우선순위를 가지게 된다.

RAMP OUTPUT = REF × (현재의 위치값)

만약 USE_REV INPUT = F, REV_RAMP= 0

만약 USE_REV INPUT =T, REV_RAMP= (P_SEL_X)(현재 RAMP값) + (P_SEL_1)

ramp의 출력이 증가하면 MOVING_UP이 T로 되고 출력이 감소하면 MOVING_DN이 true가 된다. 어떠한 기울기가 선택되어지지 않으면 기본 값이 자동으로 설정된다. 위치값과 기울기를 추가할 경우 repeat 명령을 사용한다. 입력값이 높을수록 우선순위도 높아진다. ramp가 P_LIM_X에 의해 선택된 위치이면 true가 되는데 그 위치를 벗어나게 되면 P_LIM_X는 F로 된다.

4.2.13 LSS_BUS (Low Signal Selector BUS)

이 블록은 Low signal select의 논리를 수행한다. 출력은 입력값의 제일 낮은 값을 선택하고 High limit와 Low limit 사이에 존재한다. 만약 출력값이 Low limit보다 낮으면 하한값을 내보내고, High limit보다 높으

면 상한값을 출력한다. 또한 각각의 Low limit indicator(LO_LIM_I)와 High limit indicator(HI_LIM_I)가 true로 된다.

4.2.14 ONE_SHOT

입력이 true로 되었을 때, 출력이 특정의 지연 시간 동안 true로 간다. 지연 시간이 끝나거나 RESET이 true가 되면, 입력이 여전히 true이더라도 출력은 false로 간다. 입력은 기능을 재 장전하기 위해 false로 가야하고 reset은 false가 되어야만 한다. 만약 지연시간이 끝나기 전에 입력이 false로 가면 reset INPUT(RST)이 true로 가지 않으면 지연 시간동안 출력은 true로 가게 된다. R_TIME(OUTPUT)은 출력 기능 시간이 끝나기까지 남은 시간을 나타내 준다.

4.2.15 AN_COMPARE (Analog Comparison)

이 블록에서는 만약 히스테리시스가 0이면 :

경우 1 : 출력은 (IN_1)이 (IN_2)보다 크거나 같게 되면 true가 되고,
(IN_1)이 (IN_2)보다 작으면 false가 된다.

경우 2 : 출력은 (IN_2)이 (IN_1)보다 크거나 같게 되면 false가 되고,
(IN_2)이 (IN_1)보다 작으면 true가 된다.

만약 히스테리시스가 양(POSITIVE)이면 :

경우 1 : 출력은 (IN_1)이 (IN_2 + HYSTER)보다 크거나 같게 되면 true가 되고,
(IN_1)이 (IN_2)보다 작으면 false가 된다.

경우 2 : 출력은 (IN_2)이 (IN_1)보다 크거나 같게 되면 false가 되고,
(IN_2)이 (IN_1 + HYSTER)보다 작으면 true가 된다.

만약 히스테리시스가 음(negative)이면 :

경우 3 : 출력은 (IN_1)이 (IN_2)보다 크거나 같게 되면 true가 되고,
(IN_1)이 (IN_2 + HYSTER의 절대값)보다 작으면 false가 된다.

경우 4 : 출력은 (IN_2)이 (IN_1 + HYSTER의 절대값)보다 크거나 같게 되면 false가 되고 (IN_2)이 (IN_1)보다 작으면 true가 된다.

4.2.16 PID

이 블록은 운전환경에 따라 비례(P), 비례-적분(PI) 또는 비례-적분-미분(PID) 제어 기능을 수행하게 된다. 설정값은 시스템의 출력과 비교되고 교정된 출력(0~100의 scale)이 감지된 오차로부터 출력되게 된다

$$G = P \text{ (동조시 사용)}$$

$$G = P (I / s + 1)$$

$$G = P (I / s + 1) (Ds + 1)$$

여기서,

$$D = 1 / \{ (SDR)(I) \} \text{ 혹은}$$

$$D = SDR / I$$

입력 PROP_GN은 PID블록 응답의 비례이득을 결정한다. 이는 현재 기계식 조속기의 이득의 조정과도 같다. 입력INT_GN은 PID 블록 응답의 적분이득을 결정한다. 이는 기계식 조속기의 reset의 조정과도 같다. 입력 INT+GN_MIN은 적분이득의 최소(Integral gain minimum)값으로서 블록의 기능이 비례 제어일 경우에만 사용된다. 이 PID블록이 비례 제어기의 기능만을 가지려면, 입력 PROP_ONLY이 true 이고, INT_GN의 입력값이 INT_GN_MIN의 입력값보다 작아야 한다. 입력 S_D_R은 속도 미분율(Speed derivative ratio)로서 블록의 응답이 미분요소에 영향을 주게 된다. S_D_R은 PID제어기가 요구되는 곳에서만 사용되고, 이 값은 통상 5.0으로 설정된다. S_D_R은 적어도 0이나 그 이하로 설정해서는 안된다.

$$S_D_R = 1 \sim 100 \text{ 일 때, } D = 1 / \{ (SDR)(I) \} \text{----- (케환 우선)}$$

$$S_D_R = 0.01 \sim 1 \text{ 일 때, } D = SDR / I \text{----- (입력값 우선)}$$

입력값 threshold는 이 블록의 출력이 101[%](LSS)나 1[%](HSS)로 가

기 전에 얼마만큼의 오차(설정값과 출력과의 차)가 허용되는가를 결정하는데, 이때는 이 블록이 주는 LSS bus나 HSS bus를 조정하지 않는다. 만약, threshold 입력이 0으로 세트되고, 이 블록의 출력이 LSS bus로 가고 있는데, bus의 조정을 잃게 되면, 블록의 출력은 101[%]로 가게 된다. 만약, LSS bus대신 HSS bus로 블록의 출력이 가고, threshold값이 0일 때, 이 블록이 HSS bus의 조정을 잃게 되면 블록의 출력은 1[%]로 가게 된다. 이는 threshold의 값을 0으로 세트하는 것이 바람직하지 않다는 것을 나타낸다. 최소값은 전 scale은 0.01~1.0[%]내에 있어야하고, 대부분의 모든 적용대상에서의 threshold 세트가 최대 스케일 값으로 가야한다. threshold가 0보다 크게 세트되어 있을 때, PID블록의 조정상실이 일어나도 어떤 양만큼의 오차가 존재할 때까지는 101[%]나 1[%]로 가지 않는다. threshold의 입력값은 오차의 양을 결정한다. PID블록의 출력은 아래와 같은 상태를 만나야만이 101[%]나 1[%]로 갈 수 있다.

PID블록이 LSS bus로 출력을 내고 있을 때,
 만약 실제값 < 기준값 이고,
 (실제값 - 기준값) > Threshold 이고,
 IN CONTROL 부울대수 = False 이면,
 출력은 101[%]로 가게 된다.

PID블록이 HSS bus로 출력을 내고 있을 때,
 만약 실제값 > 기준값 이고,
 (실제값 - 기준값) > Threshold 이고,
 IN CONTROL 부울대수 = false 이면,
 출력은 1[%]로 가게 된다.

PROC(Process input)은 블록으로 가는 실제의 속도, 압력들의 신호이다. SP(Set Point)는 기준입력이며, 블록으로 입력되는 시스템 입력과 같게 만드는 값이다. In control signal 부울대수 입력은 PID블록의 출력에 해당하는 SEL_1 출력이어야만 한다. 이 In control signal은 PID블록이 그 bus를 제어하게 되면 true이고, 그 bus에 대해 제어를 하지 않으면 false가 된다. PID블록이 LSS bus나 HSS bus쪽으로 신호를 보내게 되면, In control signal은 true로 가게 된다. Rate control 입력은 미분가

속이나 미분감속제어를 사용할 때 true이어야 한다. 이는 PID에서의 출력을 변형시키는 에러를 제거하는 역할을 하게 된다.

4.2.17 N_SW (ANALOG SWITCH)

이 블록은 CTRL입력이 false이면 아날로그 입력이 N.C로 들어가 출력될 수 있게 한다. CTRL 입력이 true이게 되면 아날로그 입력이 N.O로 출력된다.

CTRL = T : OUTPUT = N.O.

CTRL = F : OUTPUT = N.C.

4.2.18 _FILP FLOP

이 블록은 토글타입의 플립플롭으로 언제라도 토글 입력이 T이면 출력 또한 토글 된다.

4.2.19 DEAD BAND

이 블록의 기능은 불감대 이고 enable이 true이면 VAL과 DEMAND는 아래와 같이 수행된다.

(DEMAND VAL) > (DB_1) , RAISE_VAL = True

(VAL DEMAND) > (DB_1) , LOWER_VAL = True

(ABSOLUTE DEMAND VAL) > (DB_2) , EXCD DB_2 = True

제 5 장 속도제어 알고리즘의 동조와 시뮬레이션

5.1 제어대상의 모델링과 파라미터

5.1.1 제어대상의 모델링

알고리즘을 구성하기 위해서는 시스템의 동특성을 수학적 모델로써 나타낼 필요가 있다. 시스템의 입 출력 데이터로부터 수학적 모델을 얻는 것을 시스템 식별이라고 한다.

일반적으로 식별은 다음의 세 가지 단계를 거치게 된다^{[18],[19]}.

- 모델의 구조 결정
- 입 출력 데이터로부터 미지의(未知) 파라미터의 추정
- 모델의 검증

그러나 시스템의 식별 문제의 대부분은 모델의 구조를 결정해 두고, 그 모델에 포함되어지는 미지 파라미터를 추정하는 문제를 다루고 있는데 이를 파라미터추정이라 한다. 대부분의 시스템은 물리법칙, 원리 등에 의하여 그 주된 특성이 알려져 있다. 이러한 물리법칙에 의한 수식적 모델은 경제성이나 정밀도에 따라 수식적 모델의 구체화 또는 간략화가 가능하다. 그러나 주로 시스템 파라미터가 불변이거나 제어기 파라미터가 고정된 경우에 양호한 특성을 가진다^[20]. 식별의 주요한 절차 중^{[21],[22]} 식별하고자 하는 시스템에 따라서 개별적 현상론에 기준하여 해결하지 않으면 안 되는 경우가 많다. 본 연구에서는 제어대상의 모델의 구조를 먼저 얻고 모델의 파라미터 값은 실제 터빈의 운전 데이터를 수집하여 얻도록 한다.

그림 5.1은 본 연구에서 구성되는 발전용 증기터빈의 속도제어시스템을 보여주고 있다.

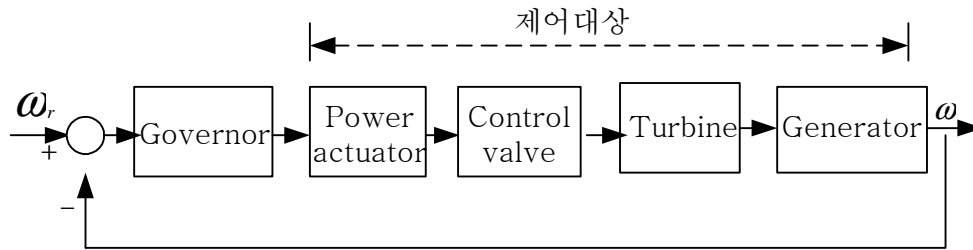


그림 5.1 발전용 증기 터빈의 속도제어 시스템

Fig. 5.1 Speed control system of steam turbine for power plant

증기 터빈 구동축으로부터 측정한 회전속도를 피드백 하여 피드백 제어 시스템을 구성하였고, 이 신호와 목표 값을 이용하여 디지털 조속기로부터 계산되어 출력되어지는 제어입력 신호는, 파워 액츄에이터에 입력되어 변위를 일으켜 제어밸브의 변위를 조절하게 되고, 이것이 증기터빈의 입력증기량을 가감하여 동작회전수를 주어진 목표 값에 추종하게 해준다. 상기의 속도제어시스템에서 액츄에이터, 제어밸브 및 터빈과 발전기를 모두 포함하여 제어대상으로 간주한다.

제어계의 설계에 있어서, 제어대상의 각 요소들이 가지는 마찰부위에서의 쿠롱마찰, 동작의 한계값 제한(액츄에이터, 제어밸브), 링크부의 비선형 원운동과 같은 비선형 요소 등을 포함하고 있어서 정확한 모델을 얻기는 어려우나 발전부의 운전조건의 변화 등을 고려한다면 제어대상을 선형 시불변인 공칭모델로 기술할 수 있고^{[23]~[27]} 선형의 불확실성을 포함한 『모델의 집합』으로 기술할 필요가 있다. 모델집합의 크기와 달성 가능한 제어 성능 간에는 밀접한 관계가 있기 때문에 공칭모델과 불확실성의 평가는 제어계 설계에 있어서 아주 중요한 단계라고 할 수 있다. 따라서 본 연구에서는 제어대상의 모델링에 있어서 타당성을 고려하도록, 몇 개의 다른 운전영역을 설정하고, 각각의 운전영역에 있어서의 응답특성의 집합으로부터 공칭모델과 불확실성을 포함한 모델을 결정한다.

정속 운전시 부하변화에도 출력이 목표값에 잘 추종하는 속도제어시스

템을 설계하기 위하여 액츄에이터 입력단에서 터빈의 회전속도까지의 제어대상을 동작점 부근에서 선형화하여 각각의 동특성을 수학적으로 표시하기로 한다.

가. 파워 액츄에이터

발전시스템에서 증기터빈의 회전수는 주 증기 제어밸브를 통해 유입되는 증기량에 의존하며, 증기량의 가감은 주 증기 제어밸브의 래크 위치를 조절 하므로써 가능하다. 일반적으로 제어밸브는 여러개의 밸브로 조합되어 있고 링크를 통해 이들의 개폐정도를 조절하기 위해서는 큰 동력의 액츄에이터가 요구된다. 북제주화력발전소의 터빈용 디지털조속기를 개발하는 본 연구를 수행하면서 이에 적합한 액츄에이터로 Woodward Governor사의 EHPC^[28]를 선정하였다. EHPC는 전기유압식 액츄에이터로서 디지털 조속기로부터 전기적 입력신호를 받아 이에 비례하는 변위신호를 출력하는데 액츄에이터의 파워피스톤은 유압동력원으로 부터 구동된다.

여러 다른 액츄에이터와 같이 EHPC 유압식 액츄에이터도 쿠롱마찰, 상하한값의 제한과 같은 비선형 요소 등을 포함하고 있으나 운전부하 부근에서 이를 선형화하면 식 (5.1)과 같이 1차계의 전달함수로 표시될 수 있다.

$$\frac{M}{U} = \frac{K_a}{1 + T_a s} \quad (5.1)$$

여기서 U와 M은 각각 액츄에이터의 입출력 신호이고 K_a 는 액츄에이터의 이득, T_a 는 액츄에이터의 시정수이다.

나. 제어밸브

본 연구의 최종목표인 터빈의 속도를 제어하기 위해서는, 터빈의 회전속도를 검출하여 목표값과 비교하고, 그 차만큼 주 증기 제어밸브를 개폐하여 증기량을 조절하여야 한다. 북 제주 화력발전소의 터빈용 주 증기 제어밸브는

5개의 밸브군으로 구성되어 있고 변위에 따라 증기량을 조절하기 위해 4-2-5-1-3 순으로 밸브가 열리고 반대로 3-1-5-2-4 순으로 닫힌다. 그림 5.2는 제어밸브의 동작 특성을, 즉 밸브 변위와 증기량과의 관계를 도시적으로 보여주고 있다. 초기 동작에서 약간의 불감대가 있으며 이후부터는 변위에 비례해서 증기량이 변화하는 것으로 모델링 할 수 있다. 여기서 K_v 는 밸브 비례상수이다.

$$Q_m = \begin{cases} K_v, & m > 5.0 \\ 0, & m < 5.0 \end{cases} \quad (5.2)$$

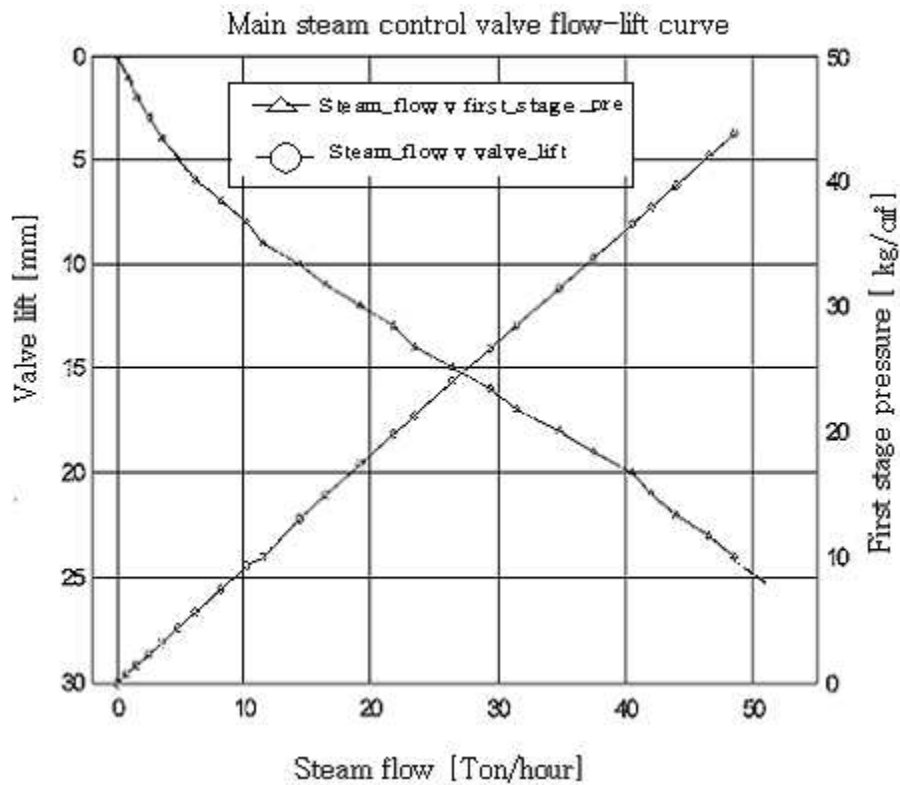


그림 5.2 제어밸브의 동작 특성

Fig. 5.2 Operating characteristic of control valve

다. 증기 터빈

증기 터빈을 모델링 행하기 위하여 이를 그림 5.3의 증기 Vessel로 간주하고, 이를 근사화 하여 전달함수를 얻고 증기 터빈 단에서 생성된 동력을 구하면 다음과 같다.

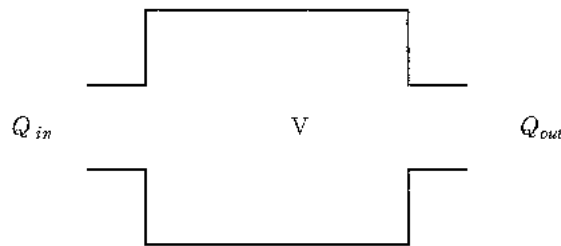


그림 5.3 증기 vessel
Fig. 5.3 Steam vessel

증기 Vessel에 대한 연속 방정식은 다음과 같이 표현된다.

$$\begin{aligned} \frac{dW}{dt} &= V \frac{d\rho}{dt} \\ &= Q_{in} - Q_{out} \end{aligned} \quad (5.3)$$

단, W = weight of steam in the vessel[kg] = $V\rho$

V = volume of vessel[m³]

ρ = density of steam[kg/m³]

Q = steam mass flow rate[kg/s]

그리고 Vessel로부터의 증기 유출량은 Vessel에서의 압력에 비례한다고 가정한다. 즉,

$$Q_{out} = \frac{Q_0}{P_0} P \quad (5.4)$$

단, P = pressure of steam in the vessel[kPa]

P_0 = rated pressure

Q_0 = rated flow out of vessel

만일 Vessel 내의 온도가 일정하다고 한다면 다음의 관계식이 성립한다.

$$\frac{d\rho}{dt} = \frac{dP}{dt} \frac{\partial\rho}{\partial P} \quad (5.5)$$

그리고 어떤 온도가 주어졌을 때 $(\partial\rho/\partial P)$ 에 대한 증기의 밀도변화는 증기표로부터 알 수 있다. 식(5.3)-(5.5)로부터

$$\begin{aligned} Q_{in} - Q_{out} &= V \frac{\partial\rho}{\partial P} \frac{dP}{dt} \\ &= V \frac{\partial\rho}{\partial P} \frac{P_0}{Q_0} \frac{dQ_{out}}{dt} \\ &= T_t \frac{dQ_{out}}{dt} \end{aligned} \quad (5.6)$$

$$\text{여기서, } T_t = \frac{P_0}{Q_0} V \frac{\partial\rho}{\partial P} \quad (5.7)$$

식(5.6)를 라플라스 변환하여 전달함수를 구하면 다음과 같다.

$$\frac{Q_{out}}{Q_{in}} = \frac{1}{1 + T_t s} \quad (5.8)$$

여기서 T_t 는 증기 Vessel의 시정수이다.

한편 증기터빈 단에서 발생된 토크, 즉 각 회전날개에 전달되는 힘은 증기 유량비에 비례한다.

$$T_m = K_t Q_{out} \quad (K_t = \text{비례상수}) \quad (5.9)$$

라. 회전계

터빈과 발전기로 구성되는 회전계의 평형방정식으로부터 기계토크 T_m 과 전기토크 T_e 의 차이가 속도의 변화를 가져오게 된다. 다음의 전달함수는 회전축의 속도와 토크사이의 관계를 나타내고 있다.

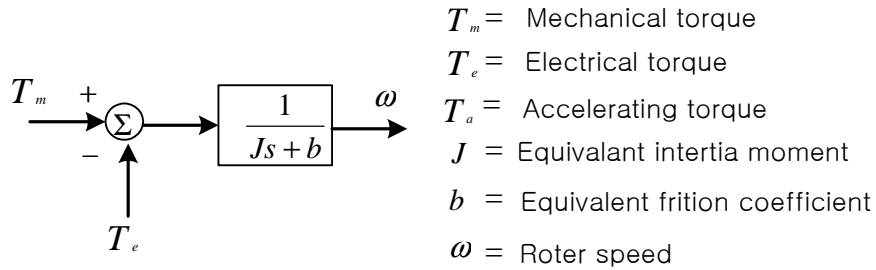


그림 5.4 속도와 토크사이의 관계

Fig. 5.4 Relation of speed and torque

이상의 각 요소들의 선형 전달함수를 결합하여 블록선도로 표시 한 것이 그림 5.5이며 제어대상의 전체 전달함수를 구하면 식(5.10)과 같다.

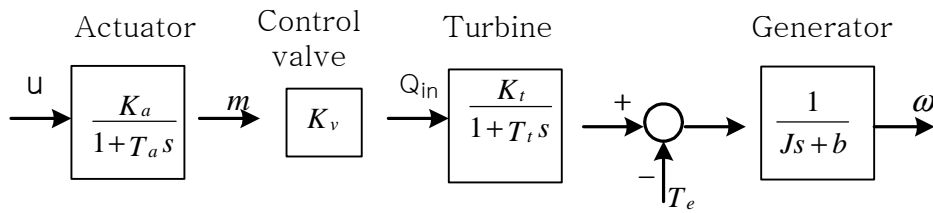


그림 5.5 제어대상의 블록선도

Fig. 5.5 Block diagram of control target

$$\Omega(s) = \frac{K_a K_v K_t}{(1 + T_a s)(1 + T_t s)(b + Js)} U(s) - \frac{1}{(b + Js)} T_e(s) \quad (5.10)$$

5.1.2 제어대상의 파라미터

5.1.1 절에서 각 계통의 전달함수를 얻었고 이로부터 제어대상의 전체 전달함수를 구할 수 있었다. 일단 모델이 결정되면 다음 단계는 관측된 입·출력 데이터를 이용하여 목적함수 값을 최적으로 하는 파라미터들을 추정하

는 것이다. 일반적으로 직면하게 되는 증기 시스템의 모든 구성들의 파라미터는 상기 블록선도에서 보여주는 파라미터와 유사하다. 본 연구에서는 미지의 파라미터를 추정하기 위해서 실험적 방법을 이용하며, 그 순서를 간단히 정리하면 다음과 같다.

1) 몇 개의 운전영역(50[%]부하, 75[%]부하, 100[%]부하)을 정하고, 각각의 운전영역에 있어서, 조속기에 적절한 입력신호를 인가하면서 액추에이터 입력과 터빈의 회전수를 입·출력 신호로 하여 데이터를 수집한다.

2) 얻어진 데이터를 주파수영역에서 해석하고 불확실한 잡음의 영향을 클 경우에는 필터를 통해 제거한다.

3) 수집한 입·출력 데이터와 적절한 추정 알고리즘(또는 적용 메커니즘)으로부터 공칭계의 파라미터를 결정한다.

매개변수를 추정할 때 상당한 크기의 잡음을 포함하는 신호를 직접 사용하는 것은 추정의 성능을 저하시키는 요인이 되므로 측정 신호로부터 잡음을 제거하기 위하여 IIR 디지털 저역통과 필터를 설계하였다. 일반적으로 디지털 필터를 설계하는 한 기법으로 먼저 차단 주파수, 통과대역 오차, 저지대역 오차 등 필터사양을 만족하는 아날로그 필터의 전달함수를 얻고, 이를 다시 디지털 필터로 변환하는 방법이 있다^[29].

실제 많은 추정문제에 있어서 잡음, 외란, 모델링 오차 등으로 그 탐색공간은 대단히 복잡하게 되어 Multi-model인 경우가 흔하다. 이 경우에 경사법(Gradient method)에 기초한 추정 알고리즘을 사용하면 전역해(Global solution)를 얻는데 어려움을 겪게 되므로 강인한 탐색방법이 요구된다.

최근 최적화 도구로서 광범위하게 이용되고 있는 유전알고리즘(Genetic algorithm)^[30]은 목적함수 외에 탐색공간에 대한 사전지식을 필요로 하지 않고, 매우 크고 복잡한 공간일지라도 전역해 쪽으로 수렴해가는 탐색능력을 가지므로 이러한 문제를 해결할 수 있다. 따라서 본 연구에서는 소규모 집단을 효율적으로 다루고 온라인 목적에 적합한 하이브리드 유전알고리즘(Hybrid Genetic Algorithm: HGA)^[31]을 적응 메커니즘으로 이용한다. 그림 5.6은 하이브리드 유전알고리즘의 동작 과정을 보여 주고 있다.

계측된 입력을 모델에 인가하였고, 하이브리드 유전알고리즘은 시스템

의 출력과 모델의 출력이 최소가 되도록, 즉 모델의 동적특성이 시스템의 동적특성에 가까워지도록 계속적으로 모델의 파라미터를 조정하게 된다.

PROCEDURE OF THE HYBRID GENETIC ALGORITHM
 Set $k=0$
 Create an initial population randomly;
 Decode individuals into parameter vectors;
 Evaluate fitness incorporating the scaling window scheme and determine the best;
 WHILE \langle the termination conditions are not met \rangle
 Assign a new vector to each individual using the gradient-like formula;
 Encode the now vectors into structures to form mating pool;
 Crossover and mutate;
 Decode individuals into parameter vectors;
 Evaluate fitness incorporating the scaling window scheme
 and determine the best(output the potential solution at this step);
 Set $k=k+1$
 END WHILE

그림 5.6 하이브리드 유전알고리즘의 동작과정

Fig. 5.6 Process of H.G.A.

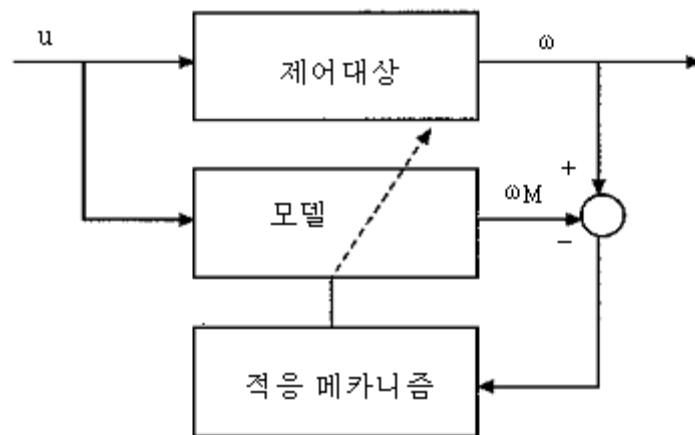


그림 5.7 제어대상 시스템의 파라미터 추정 구성도

Fig. 5.7 Diagram of parameter for control target

개체집단이 진화해 가는 동안 하이브리드 유전알고리즘은 개체간의 우열, 즉 각 개체로부터 복호화된 파라미터로 설정된 모델의 성능을 평가하기 위하여 목적함수를 필요로 하는데 적합도는 이 목적함수로부터 적절히 계산되어진다. 온라인 추정값을 얻기 위하여 목적함수는 유한시간 동안만 계산되는 다음과 같은 식을 이용하였다.

$$J(\psi) = \sum_{j=k-W+1}^k |\omega(j) - \omega_M(j)| \quad (5.11)$$

여기서 $\psi = [K_a, K_v, K_t, T_a, T_t, J, b]^T$ 는 조정되는 파라미터 벡터이고, W는 데이터 윈도우의 크기이다. W의 크기는 추정값의 정확도와 연산시간 사이에서 적절히 타협되어 선택된다. 하이브리드 유전알고리즘은 목적함수의 값이 최소가 되도록 ψ 를 조정해 나간다.

5.2 조속기의 설계 알고리즘

5.2.1 설계사양

증기 터빈을 조속하기 위한 디지털 조속기의 기본적인 기능은 다음과 같이 4가지로 요약할 수 있다^{[32]~[34]}.

- 1) normal speed/load control
- 2) overspeed control
- 3) overspeed trip
- 4) start-up/shutdown control

이외에 다수의 부수적인 기능들이 추가될 수 있다. Speed/load 제어는 액츄에이터 입력신호를 제어함으로써 이루어지며, 특히 속도제어를 행하는데 있어서는, 시스템간의 적절한 부하분담으로 만족스러운 수준에서의 목적달성을 위해서 4~5[%]의 스피드드롭까지 고려하고 있으며, Speed/load 제

어는 Speed/load 기준입력 (또는 설정값)을 조정함으로써 이루어진다. 발전 시스템은 상당한 시간동안 Speed/load 제어 모드에서 동작되기 때문에 먼저 Speed/load 제어에 필요한 알고리즘을 개발한다. 지금까지의 내용을 바탕으로 해서, Speed/load 제어하도록 그림 5.8과 같은 구조를 갖는 제어계를 구성한다.

발전용 증기 터빈의 동특성은 운전조건이나 출력조건 등의 운용조건에 크게 의존하는 복잡한 열역학적 플랜트이다. 발전 중 부하변동 등으로 인한 운전조건에 따른 제어성능의 열화 등, 최적운전을 위한 요건들을 충분히 고려하여 만족되는 제어계를 설계하기 위하여 우선 제어대상의 모델을 얻는 것이 필수적이다. 특히 발전시스템은 분담할 부하의 크기에 따라 시스템의 파라미터 변화가 극심하므로 전 운전조건에서 양호한 모델을 얻는다는 것은 상당한 어려움이 있다. 따라서 본 연구에서는 자주 운전되는 부하점에서 제어대상의 모델을 선형화하였으며 각각의 동작속도에서 주어진 응답특성을 만족하는 PID제어를 행하고 조속되도록 한다.

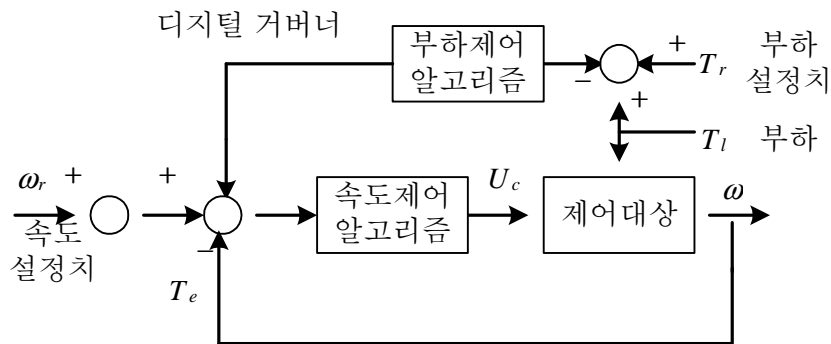


그림 5.8 속도/부하 제어 알고리즘을 갖는 제어계
 Fig. 5.8 Control system of SPEED/LOAD algorithm

5.2.2 PID 제어 알고리즘 도출

사용목적에 따라 3가지 동작 중 그 일부를 조합한 동작을 P제어, PI 제어, PD 제어라고 하며, 모두를 포함할 때는 PID 제어라 한다. 이러한 동작을 하드웨어 또는 소프트웨어적으로 구현한 장치를 P 제어기, PI제어기, PD제어기 또는 PID제어기라 한다.

목표값에서 관측량을 뺀 값을 편차(또는 오차) e 로서 나타내기로 한다면 이 편차에 따라 PID 제어기는 조작량을 결정하게 된다. PID 제어의 제어칙은 시간영역에서 식(5.12)로 나타내어진다.

$$u(t) = K_P \left\{ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right\} \quad (5.12)$$

식 (5.12)의 PID 제어동작을 주파수영역으로 표시하면 식(5.13)과 같다.

$$U(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) E(s) \quad (5.13)$$

그림 5.9는 PID 동작의 한 예를 보여주고 있다

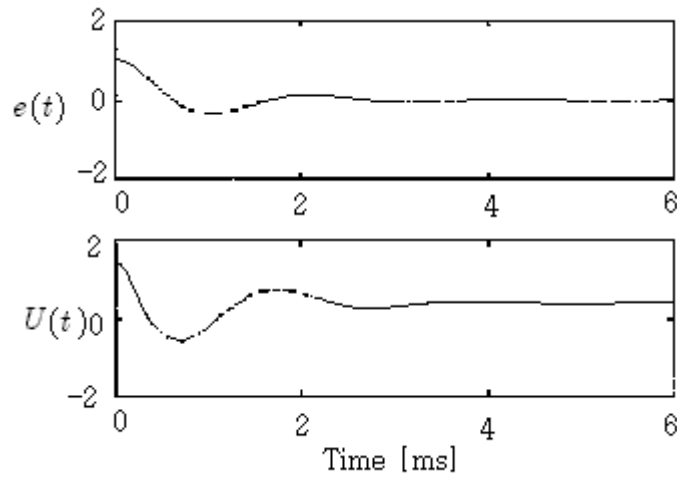


그림 5.9 PID 제어

Fig. 5.9 PID control

5.2.3 PID 파라미터의 오프라인 동조

PID 제어기의 파라미터, 즉 비례이득, 적분시간, 미분시간의 적절한 값을 선택하기 위한 지침을 PID 제어기의 동조(tuning)라 한다. 이것을 크게 개루프의 스텝 응답특성과 폐루프의 주파수 응답특성에 의한 방법으로 분류할 수 있다.

가. 스텝 응답특성에 의한 방법^[35].

개루프 상태에서 제어대상에 단위 스텝함수의 입력을 가해 제어대상의 스텝응답을 구한다. 대부분의 프로세스는 식 (5.14)와 같이 1차 늦음과 시간지연의 결합으로 근사화 할 수 있으므로 스텝응답으로부터 K, T, L 을 얻는다. 스텝응답은 일반적으로 시그모이드형의 곡선으로 나타난다. 이것을 프로세스 반응곡선이라 한다. 이 곡선에서 가장 기울기가 급한 부분, 즉 변곡점에서 접선을 긋고 이때의 기울기를 R이라고 한다. 이것을 반응속도라고도 부른다. 이접선이 횡축과 만나는 시각을 L 이라 하고, 이것을 늦음(delay)이라고 부른다. 제어량이 최종적으로 정착될 때의 값을 정상이득이라 하고 이것을 K로 둔다.

$$G(s) = \frac{K}{1 + Ts} e^{-Ls} \quad (5.14)$$

결국, 이러한 3개의 파라미터, 혹은 L을 제외한 2개의 파라미터를 가지고 PID 제어의 계수를 결정하게 된다. Ziegler와 Nichols는 실험 예로부터, PID 제어의 파라미터를 R, L 과 관련지어서 결정하는 조정칙을 제안하였다. 그것은 표 5.1과 같다.

표 5.1 Ziegler-Nichols의 스텝응답을 이용한 동조

Table. 5.1 Tuning for step response of Ziegler-Nichols

제어기 형태	제어기 파라미터		
	K_P	T_1	T_D
P	1/RL	-	-
PI	0.9/RL	L/0.3	-
PID	1.2/RL	2L	0.5L

나. 1/4 감쇠법

이는 폐루프 상태에서 비례동작만으로 제어를 행하고, 제어계의 감쇠 특성에 관한 정보를 이용하여 조정하는 방법이다. 1/4감쇠라고 하는 것은, 최초의 오버슈트량과 그다음의 오버슈트량에 대해 다음의 관계가 성립될 때를 말한다.

$$\frac{\text{최초의오버슈트량}}{\text{두번째의오버슈트량}} = \frac{1}{4} \quad (5.15)$$

이것은 감쇠특성의 기준이 되며, 이러한 관계가 성립되는 것이 바람직하다고 알려져 있다. 실제의 제어대상을 안정 한계까지 이끌고 가서 지속적인 진동을 발생시키는 것은 그다지 바람직 한 것이라 할 수 없다. 따라서 안정한계 대신 위에서 기술한 1/4 감쇠진동 상태에 착안하는 조정책이 제안되어져 있다. 이 방법은 비례제어에서 비례이득을 조정하여 1/4 감쇠진동 상태까지 이끌고 가는 방법이다.

그때의 진동 주기를 $T_{1/4}$ 이라고 한다. PID 제어에 있어서의 적분시간, 미분시간을 다음과 같이 선택하고, 비례이득은 1/4 감쇄진동 상태가 되도록 조정하면 된다.

$$T_I = \frac{T_{1/4}}{1.5} \quad (5.16)$$

$$T_D = \frac{T_{1/4}}{6} \quad (5.17)$$

다. 한계감도법^[35]

이 방법 역시 페루프 상태에서 비례 동작만으로 제어를 행하고, 제어계의 주파수 특성에 관한 정보를 이용하여 조정되는 방법이다. 비례 제어로 비레이득을 조금씩 증가시켜 가면, 제어량의 응답은 점점 진동적으로 되어 결국은 안정한계를 넘어 발진상태로 되는 것이 보통이다. 임계점 즉, 일정진폭의 지속적인 진동이 계속되는 상태로 될 때의 비레이득을 임계이득 K_c 라 하고, 그때의 주기를 임계주기 T_c 라 한다. 제어대상의 주파수 응답과 관련지어서 기술한다면, $\angle G(j\omega) = -180^\circ$ 로 되는 각 주파수를 ω_1 이라고 할 때 $T_c = 2\pi/\omega_1, K_c = 1/|G(j\omega_1)|$ 로 된다. Ziegler와 Nichols는 실험 예를 통해서, PID 제어의 파라미터를 $K_c T_c$ 와 관련지어서 표 5.2와 같이 결정하도록 하고 있다. P제어의 비레이득은 1/4 감쇠가 되도록, 그 외의 파라미터는 그와 거의 동등한 감쇠특성이 얻어질 수 있도록 결정한 것이다. PID 매개변수는 어디까지나 경험적 방법으로 동조된 결과이므로 설계사양에 부합하면서 주어진 시스템에서 성능이 개선되도록 미세 조정될 필요가 있다^[36].

표 5.2 한계감도법에 의한 동조

Table. 5.2 Tuning for marginal sensitivity

제어기 형태	제어기 파라미터		
	K_P	T_I	T_D
P	$0.5 K_c$	-	-
PI	$0.45 K_c$	$0.833 T_c$	-
PID	$0.6 K_c$	$0.5 T_c$	$0.125 T_c$

5.2.4 PID 파라미터의 자동 동조

이상에서 살펴본 조정칙들은 경험적인 과정을 거쳐서 만들어진 결과라는 것만을 감안하더라도 설계를 행하는 설계자의 입장에서 생각해 본다면, 무언가 선명하지 못한 부분들이 너무 많다. 그리고 설계에 있어서 상당한 제약조건이 수반됨을 알 수 있다. 예를 들어

- 1) 일반적으로 스텝 응답곡선으로부터 세 파라미터를 얻기가 쉽지 않다.
- 2) 임계이득과 주파수를 얻기 위하여 시스템이 임계점에서 장시간 운전되어야 하므로 불안정한 운전으로부터 시스템이 손상을 받을 수가 있다.

이런 문제점을 보완하면서 불확실한 시스템의 PID 제어기의 이득 결정 방법에는 릴레이 입력에 대한 응답을 이용하는 방법^[37], 출력 패턴을 이용한 전문가 조정 방법^[38] 등이 있다.

연구에서는 릴레이 입력에 대한 응답을 이용하여 제어기의 파라미터를 구하는 방법에 대해 살펴본다.

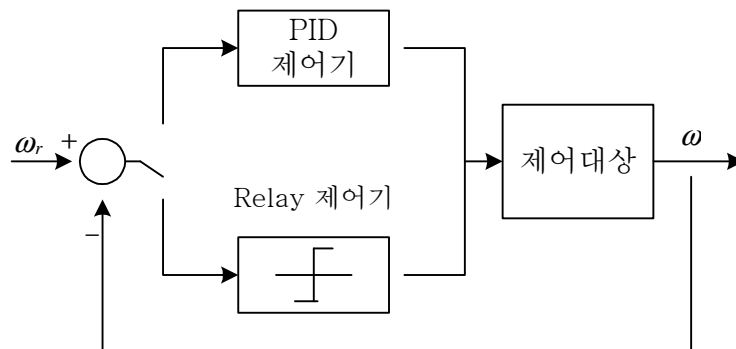


그림 5.10 Relay 피드백 제어 시스템

Fig. 5.10 Control system of relay feedback

Ziegler-Nichols법의 단점을 보완하기 위하여 제시된 릴레이 피드백에

의한 이득 결정방법의 기본개념은 대부분의 시스템은 릴레이 피드백 상황에서 그 출력이 임계주기를 가지고 진동하므로 이로부터 제어기 동조에 필요한 한계이득 (K_c)과 한계주기 (T_c)를 구할 수가 있다^{[39],[40]}.

기준입력과 플랜트 출력사이의 오차를 $e(t) = \omega_\gamma(t) - \omega(t)$ 라 하면 릴레이 제어기는 다음과 같이 동작한다.

$$u(t) = \begin{cases} +d, & e(t) > 0 \\ -d, & e(t) < 0 \end{cases} \quad (5.18)$$

여기서 d 는 릴레이 제어기의 출력크기이다. 이때 시스템이 저역 필터의 특성을 가지고 있는 경우에는 오차신호의 크기는 근사적으로 식 (5.19)와 같게 된다.

$$a = \frac{4d}{\pi} |G(j2\pi f_c)| \quad (5.19)$$

여기서 $f_c = \frac{1}{T_c}$ 는 임계주파수이다. 임계주기를 갖는 진동이 발생할 조건은 식 (5.20),(5.21)과 같다.

$$\arg G(j2\pi f_c) = -\pi \quad (5.20)$$

$$K_c = \frac{4d}{\pi a} = \frac{1}{|G(j2\pi f_c)|} \quad (5.21)$$

이 방법으로 K_c 와 T_c 가 구해지면 Ziegler-Nichols의 한계감도법에 의한 동조치를 적용할 수 있다.

5.3 시뮬레이션 및 고찰

여기서는 전 절에서 개발된 제어 알고리즘을 이용하여, 설계되어진 속도제어시스템의 성능을 컴퓨터 시뮬레이션을 통해 확인해 보도록 한다. 시뮬레이션에 사용된 제어기와 제어대상의 파라미터는 다음과 같다.

표 5.3 시뮬레이션 파라미터

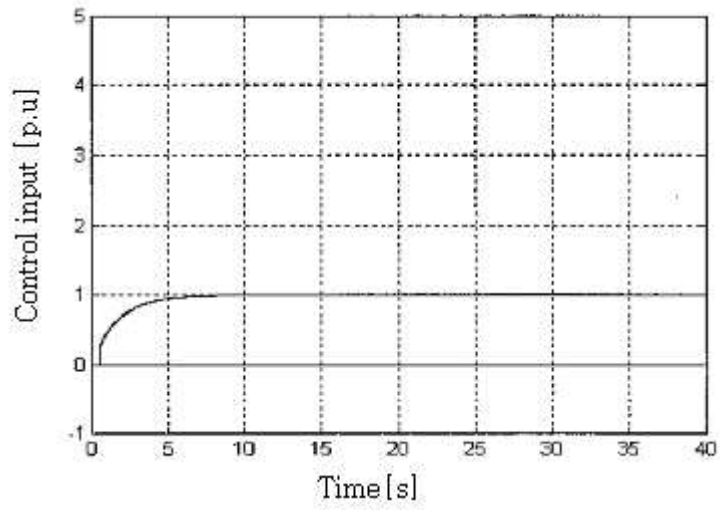
Table 5.3 Parameters for simulation

구분	파라미터
제어기	$K_p = 0.55, T_I = 0.5, T_D = 0.125$
제어대상	$K_a=1.0, K_v=1.0, K_t=1.0, Ta=0.21$

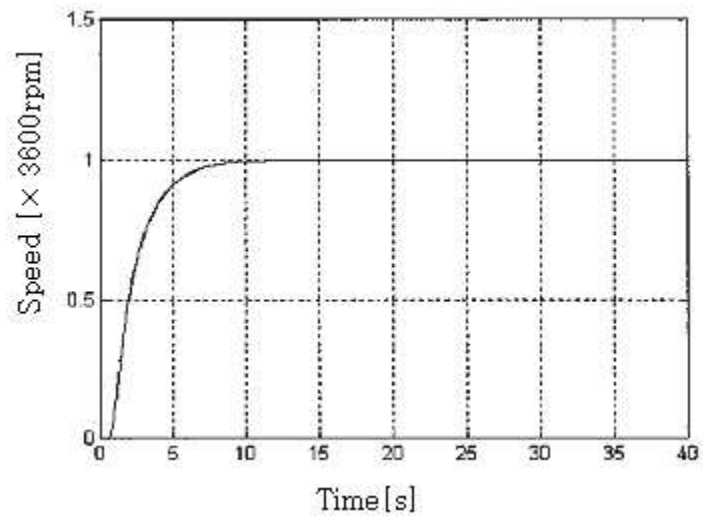
5) 제어기의 유효성을 확인하기 위해 상기 파라미터를 이용하여 MatLab simulink로 시뮬레이션을 하였다. 그래프상의 측정부위는 그림5.8에서 제어 신호인 U_C 가 Control input이다. Controlled output은 출력측의 ω 이다.

6) 그림 5.11는 스텝상의 목표신호에 대한 출력(터빈의 회전속도)응답과, 제어신호를 나타내고 있다. 조속기에 Initial value=0, Final value=1로 스텝신호를 인가한 결과 출력(터빈의 회전속도)응답신호는 제어신호와 맞춰 양호하게 응답신호가 10[s]후 정상 회복됨을 알 수 있다. 그림 5.12는 부하변동 등의 경우를 고려하여 가동 20[s]후에 제어계의 출력 측에 외란을 가한 경우는 제어계의 출력측에 일시적으로 부하가 걸려 출력 [rpm]이 정격속도의 8[%]정도 떨어진 후 3[s]지난 후 안정됨을 확인할 수 있었다. 또한 그림 5.13은 부하감쇄계수(b)가 1에서 1.2로 변동하였을 때의 응답 특성을 나타내고 있다. 부하감쇄계수를 1.0에서 1.2로 변경한 경우는 정상 출력이 15[s]수준으로 부하감쇄계수가 1.0보다는 5[s]정도 지연됨을 알 수 있다. 이는 부하감쇄계수가 커짐에 따라 응답시간이 점차 느려지기 때문인 것으로 사료된다. 그림 5.14는 외란과 부하감쇄계수의 변동을 동시에 고려한 경우의 응답특성을 나타내고 있다. 부하감쇄계수를 1.2로하고 가동

20[s]후 외란을 가한 경우의 출력측 [rpm]은 15[s]후 정상출력이 되고 가동 20[s]후 일시적인 변동은 있지만 정상출력이 됨을 알 수 있다. 이상의 결과로 설계되어진 제어기는 출력이 목표값에 잘 추종되고, 운전 중 외란을 주어도 요구조건을 잘 만족하고 있음을 알 수 있다.

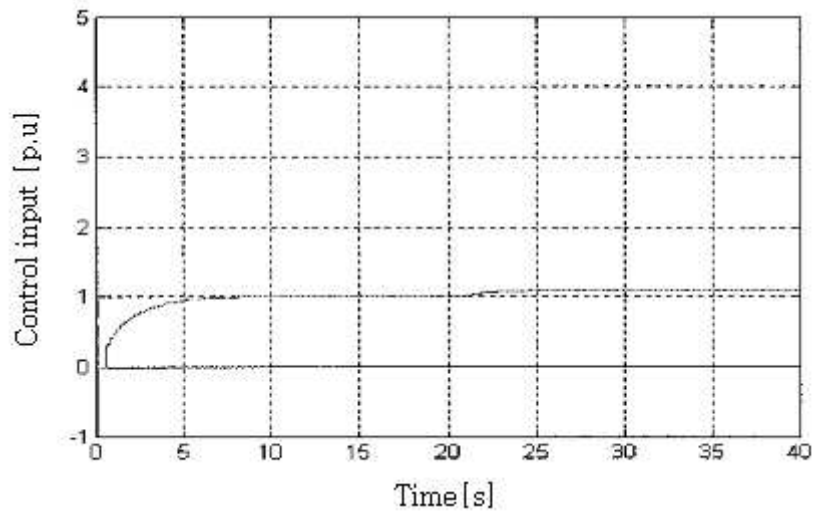


(a) Control input

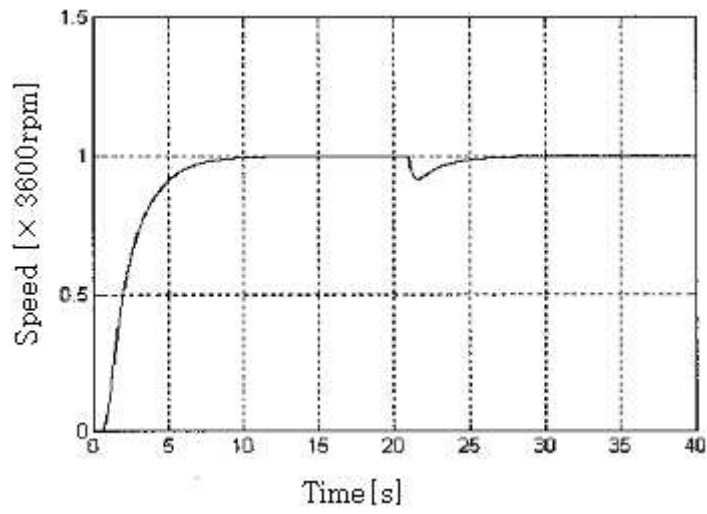


(b) Controlled output

그림 5.11 스텝응답
Fig. 5.11 Step response



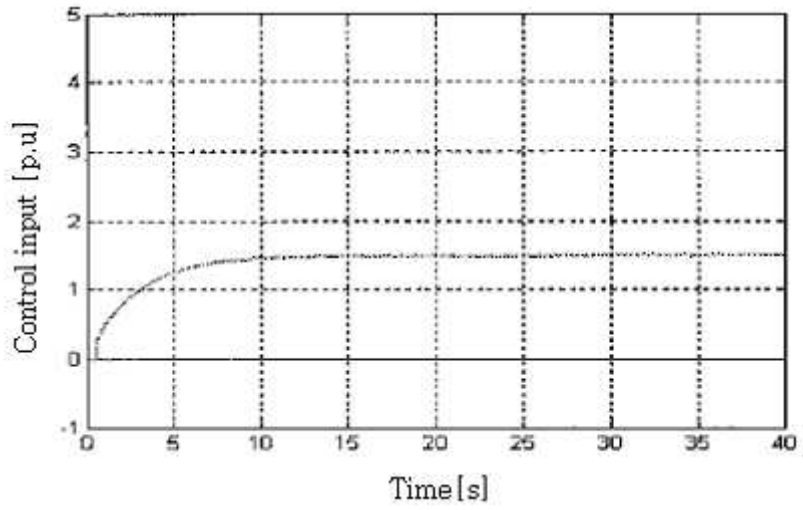
(a) Control input



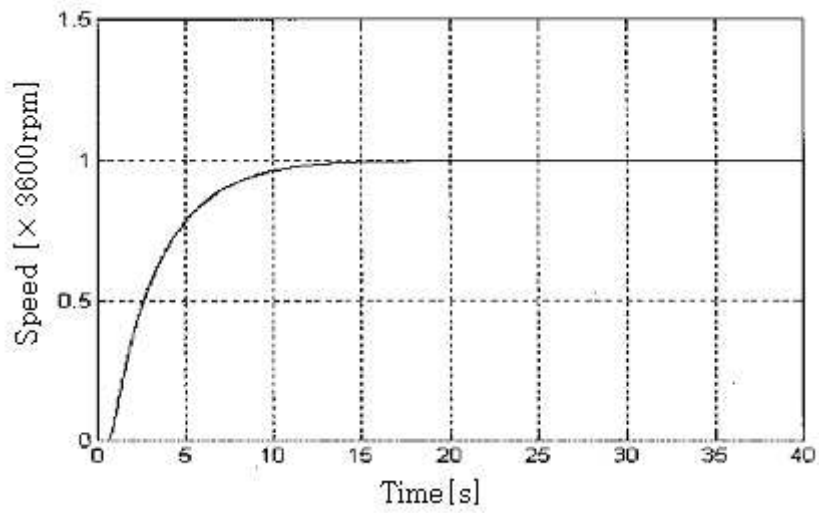
(b) Controlled output

그림 5.12 스텝상의 외란을 가함

Fig. 5.12 Step response to the perturbation



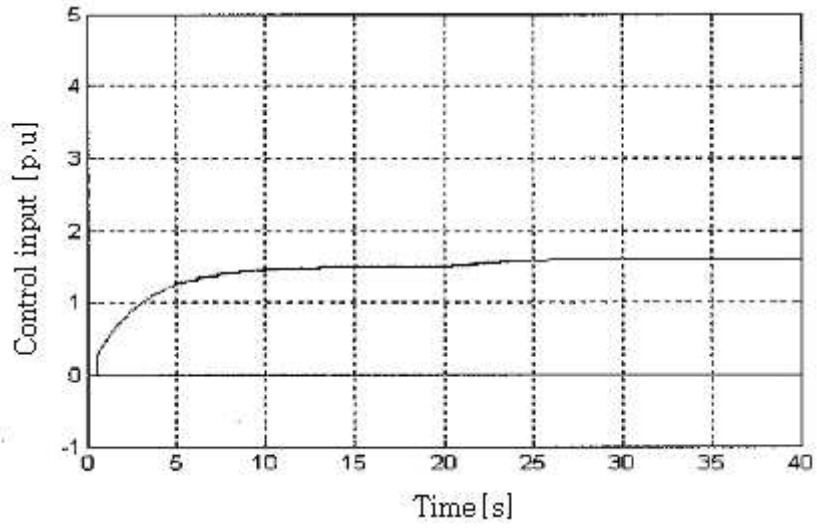
(a) Control input



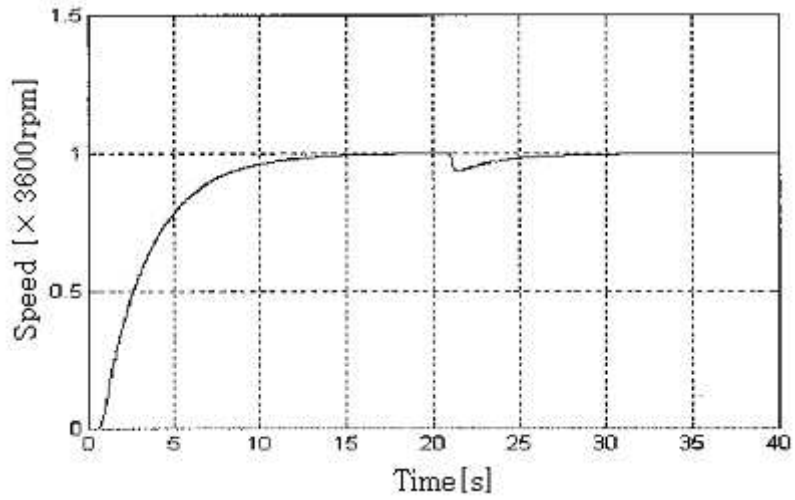
(b) Controlled output

그림 5.13 부하감쇄계수 변동시 응답특성

Fig. 5.13 Step response to the load-damping constant



(a) Control input



(b) Controlled output

그림 5.14 외란과 부하감쇄계수 변동을 고려한 응답특성

Fig. 5.14 Step response to the perturbation and load-damping constant

제 6 장 결론

본 논문은 증기터빈용 디지털 조속기의 설계에 관한 연구로서 발전기의 터빈을 컨트롤하는 제어시스템을 디지털식으로 개발하기 위한 다음과 같은 결론을 얻었다.

1) 디지털 조속기의 하드웨어는 개별메모리를 갖고 있는 CPU의 3중화와 I/O의 2중화로 내고장성을 향상시켰고 3개의 CPU 모듈은 개별 메모리를 갖고 있으며 외부 센서로부터 입력된 신호를 각각 연산처리 후 데이터를 2 out of 3 voting 을 사용하여 안정된 출력신호로 서보밸브를 구동할 수 있다.

2) 디지털 조속기의 소프트웨어는, CPU 기동에 필요한 운영시스템 프로그램과 응용프로그램으로 되어있고 각각의 기능에 맞추어 설계를 하였다.

3) 시스템 알고리즘의 개발은 Woodward Governor사의 디지털 조속기용 시스템 알고리즘을 분석하고 이를 수정 보완하여 북제주화력발전소용 디지털 조속기가 가져야할 기능들을 열거하고 시스템 알고리즘을 개발하였다. 전체시스템은 10개의 복합 블록과 이를 보조하는 19개의 기능블록으로 구성되어 있다. 복합 블록기능은 신호흐름과 그 블록이 가지는 기능이고 단일 블록은 기능들이 실제 CPU 모듈내의 메모리에 들어가게 되는 기능 프로그램으로서 외부로부터 입력되는 신호들을 어떤 방식으로 처리할 것인가를 나타내고 있다.

4) HGA를 사용해서 관측한 대상의 파라미터를 구한 결과 $K_p=0.55$, $T_i=0.5$, $T_D=1.125$ 이고 $K_a=1.0$, $K_v=1.0$, $K_f=1.0$, $Ta=0.21$, $T_f=0.3$, $J=0.24$.

$b=1.0$ 이다.

5) 제어기의 유효성을 확인하기 위해 상기 파라미터를 이용하여 MatLab simulink로 시뮬레이션을 하였다. 조속기에 Initial value=0, Final value=1로 스텝신호를 인가한 결과 출력(터빈의 회전속도)응답신호는 제어신호와 맞춰 양호하게 응답신호가 10[s]후 정상 회복됨을 알 수 있다. 가동 20[s]후에 제어계의 출력 측에 외란을 가한 경우는 제어계의 출력측에 일시적으로 부하가 걸려 출력 [rpm]이 정격속도의 8[%]정도 떨어진 후 3[s]지난 후 안정됨을 확인할 수 있었다. 또한 부하감쇄계수를 1.0에서 1.2로 변경한 경우는 정상출력이 15[s]수준으로, 부하감쇄계수가 1.0보다는 5[s]정도 지연됨을 알 수 있다. 이는 부하감쇄계수가 커짐에 따라 응답시간이 점차 느려지기 때문인 것으로 사료된다. 부하감쇄계수를 1.2로하고 가동 20[s]후 외란을 가한 경우의 출력측 [rpm]은 15[s]후 정상출력이 되고 가동 20[s]후 일시적인 변동은 있지만 정상출력이 됨을 알 수 있다. 이상의 결과로 설계되어진 제어기는 출력이 목표값에 잘 추종되고, 운전 중 외란을 주어도 요구조건을 잘 만족하고 있음을 알 수 있다.

참고문헌

- [1] 한국에너지 협의회편, 에너지협의회보, 제4차 장기 전력수급계획: 1998년부터 2015년까지, 47('98.10), pp.13-29, 1998
- [2] Rene J Bender, Power Generation Systems, McGraw Hill Book Company, 1967
- [3] Governing Fundamentals, manual No. TA-442, Woodward Governor Company
- [4] 정태환, Control of Prime Mover Speed, 한국 전력공사 정비기획실, 1997.
- [5] Ron Platz, Steam and Gas Turbine Control Retrofits, Woodward Governor Company Tech Talk No. 83407
- [6] D. G. Luengberger, "Observing the State of a linear System," IEEE Trans. On Military Electronics," Vol. MIL-8, pp.74-80, 1964
- [7] D. T. Pham and D. Karaboga, "Optimum Design of Fuzzy Logic Controllers Using Genetic Algorithms," J. Syst. Engng., Vol. 1, pp.144-148, 1991
- [8] W. Hwang and W. E. Thompson, "Design of Intelligent Fuzzy Logic Controllers Using Genetic Algorithms," Proc. 3rd IEEE Int. Conf. on Fuzzy Systems, Vol. 3, No. 2, pp.1383-1389, 1994
- [9] D. T. Pham and G. Jin, "Evolutionary Design of an Adaptive Fuzzy Logic Controllers for Processes with Time Delays," Proc. '94, IEEE Int. Conf. on SMC, San Antonio, TX, Vol. 1, pp.431-436, 1994
- [10] 한국 전력공사, 주파수조정 운전을 위한 터빈 조속기 및 보일러 제어계의 성능개선, 전력연구원 자동제어 연구실, 1987

- [11] Dave Augustine, Advanced Software System Lets Control Engineers Concentrate on Control, Woodward Governor Company Tech Talk No. 83405
- [12] Governing Fundamentals, Manual No. TA-442, Woodward Governor Company
- [13] Net Con[®] DIGITAL CONTROL Hardware Manual, Manual no. 85544E, Woodward Governor Company.
- [14] 509 DIGITAL ONTROL SYSTEM Hardware Manual, Manual No. 85512E, Woodward Governor Company
- [15] 신동명의외2인, 도면해독 기계요소제도, pp.260-262, 2002
- [16] 김동우외3인, 기계공학개론, 원창출판사, pp.189-190, 1999
- [17] 조성철외2인, 일반기계공학, 미전사이언스, pp.94-100, 2002
- [18] J G Ziegler and N. B. Nichols, "Optimum settings for Automatic Controllers," Trans. ASME, Vol.64, No.11, pp.759-768, 1942
- [19] P. Young, "Parameter Estimation for Continuous-Time Model -A Survey," Automatica. Vol.17, No.1, pp.23-39, 1981
- [20] 김성덕, 제어시스템 해석, 북두출판사,1999
- [21] D. S. Bayard, "An algorithm for state frequency domain identification without windowing distortions", Proceedings of the CDC, 1992.
- [22] C. W. Chen, J. N. Juang, & G. Lee, "Frequency domain state-space system identification", NASA TM-107659, June 1992.
- [23] 김희철, 蒸氣 터빈, 一中社, 1979
- [24] 정태환, Control of Prime Mover Speed, 한국전력공사 정비기획실, 1997
- [25] A. W. Ordys, A. W. Pike, M. A. Johnson, R. M. Katebi and M. J. Grimble, Modelling and Simulation of Power Generation Plants, Springer-Verlag, 1994
- [26] 천행춘, “중속디젤 기관의 디지털 가바나 설계를 위한 툴 개발에

- 관한 연구”, 한국해양대학교, 석사학위논문, 1997.
- [27] Woodward, Electrohydraulic Power Cylinder(EHPC)-Installation, operation and maintenance, 1993
- [28] Ordys, A.W.,A.W. Pike, M. A. Jhson, R. M. Katebi and M. J. Grimble, Modelling and Simulation of Power Generation Plants, Springer-Verlag,1994
- [29] M. S. Ghausi and K. R. Laker, Modern Filter Design, Prentice-Hall Inc., London, 1981
- [30] J. H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Michigan, 1975
- [31] D. T. Pham and G. Jin, "A Hybrid Genetic Algorithm," Proc. 3rd world Cong. on Expert Systems, Seoul, Korea, Vol. 2, pp.748-757, 1996
- [32] Woodward, The Meeting Monute on the Application of Woodward Governor to the B&W Engine, 1984
- [33] Norcontrol, Digital Governor DGS8800 System Information, 1985
- [34] S.T.Lynso, Electronic Governor System EGS900 C1900.166.6 23, ver.2.01, 1987
- [35] I. D. Landau, System Identification and Control Design, rentice-Hall International, Inc, 1990
- [36] Larry Mcleister PID Control, Woodward Governor Company, Tech Talk No.83402.
- [37] K. J. Astrom and T. Hagglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," Automatica, Vol.20, No.5, pp.645-651, 1984
- [38] 박귀태외 3인, “패턴 인식에 의한 새로운 자동조정 PI 제어기,” 대한전기학회지, 제 40권, 제 7호, pp.696-705, 1991
- [39] 권영훈, PID제어기를 이용한 발전용 스팀터빈의 조속 알고리즘

및 디지털화에 관한 고찰, 한국해양대학교, 석사학위논문, 1998.

- [40] 이찬주, 김응석, “발전소에 도입된 PID 제어기의 고찰 및 분석,”
‘97 대한전기학회 창립 50주년 하계학술대회 논문집, pp.648-650, 1997

부 록

<프로그램 1. 어드레스 디코더 PLD 프로그램>

```
"Input
  BCLK                                PIN  1;
  A31,A30,A29,A28,A27,A26,A25,A24    PIN  2,3,4,5,6,7,8,9;
  TS,TA                                PIN  10,11;
"Output
DRAM,IRAM,BRAM,ROM,IO                PIN  23,22,21,20,19 istype
REG,INVERT ;
BBCLK                                PIN  14 istype
'COM';
Declarations
ADDRESS = [A31...A24];
OUT = [DRAM,IRAM,BRAM,ROM,IO];
Equations
ROM    := !((ADDRESS==^h00) & !(TS&ROM) &TA);
DRAM   := !((ADDRESS==^h02) & !(TS&DRAM) &TA);
IRAM   := !((ADDRESS==^h03) & !(TS&IRAM) &TA);
BRAM   := !((ADDRESS==^h04) & !(TS&BRAM) &TA);
IO     := !((ADDRESS==^h05) & !(TS&IO) &TA);
BBCLK  := !(BCLK)
```

<프로그램 2. WAIT GENERATOR PLD LOGIC 프로그램>

```
"Input
```

```

    BCLK,RAMD,RAMI,RAMB,ROM,IO,TA,TS    pin 1,3,4,5,6,7,8,9
"Output
    BTA,TEA                Pin 23,22    istype `REG,INVERT`;
    Count0.Count1.Count2,Count3,Count4,  Pin 17,18,19,20,21istype
`REG,INVERT`;
STAT                        Pin 14      istype `REG,INVERT`;
Declarations
    Cnode0 = Cnodt0 # Cnodt1;
    Cnode1 = Cnodt2 # Cnod0;
    Cnode2 = Cnodt3 # Cnode1;
    Cheader = [Count4.Count0];
Equations
STAT    :=(TS&STAT) # !TA#!TEA;
Count0  :=(!Count0 $ STAT)#(!TA#TEA);
Count1  :=(!Count1 $ Count0)#(!TA#TEA);
Count2  :=(!Count2 $ Cnode0)#(!TA#TEA);
Count3  :=(!Count3 $ Cnode1)#(!TA#TEA);
Count4  :=(!Count4 $ Cnode2)#(!TA#TEA);
BTA     :=((RAMD & RAMI) # Count0)
        & (Count2 # RAMB)
        & (IO # Count2 # Count3)
        &(ROM # Count0 # Count2)) # !TA;# !TEA;
TEA     := (Count3 # Count4) # !TA # !TEA;

```

<프로그램 3. 다이내믹 사이즈 제어 PLD 로직 프로그램>

```

"Input
    SIZE0,SIZE1,A0,A1,RAMD,RAMI        pin 2,3,4,5,6,7;

```



```

"Output
    RAMD_UUD,RAMD_UMD,RAMD_LMD,RAMD_LLD    pin19,18,17,16
istype `COM`;
    RAMI_UUD,RAMI_UMD,RAMI_LMD,RAMI_LLD    pin 15,14,13,12
istype `COM`;
Declartions
    RAMD_OUT =[RAMD_UUD,RAMD_UMD,RAMD_LMD,RAMD_LLD];
    RAMI_OUT  =RAMI_UUD,RAMI_UMD,RAMI_LMD,RAMI_LLD];
Equations
    RAMD_UUD= !((!A0&!A1)#(!SIZE1&!SIZE0)#(SIZE1&SIZE0)#RAMD;
    RAMD_UMD=!((A0&!A1)#(!A1&SIZE1)#(SIZE&SIZE0)
                #(!SIZE&!SIZE0))#RAMD;
    RAMD_LMD=!((!A0&A1)#(!SIZE1&!SIZE0)#(!SIZE1&!SIZE0))#RAMD;
    RAMD_LLD=!((A0&A1)#(A1&SIZE1)#(SIZE&SIZE0)
                # (!SIZE&!SIZE0))#RAMI;
    RAMI_LMD=!(!A0&A1)#(SIZE1&SIZE0)#(!SIZE1&!SIZE0))#RAMI;
    RAMI_LLD=!((A0$A1)#(A1&SIZE1)#(SIZE1&SIZE0)
                #(!SIZE&!SIZE0))# RAMI;

```

<프로그램 4. 클록 동기 PLD 프로그램>

```

"Input
    CLK,A_CLK,B_CLK,C_CLK,A0,A1.ENA    pin 1,2,3,4,5,6,7;
"Output
    CLK_OUT,BENA                        pin 19,16  istype `COM`;
    CS_REG0,CS_REG1                    pin 18,17  istype `REG,INVERT`;

```

Equations

```
CLK_OUT=!((CS_REG0&!CS_REG1&A_CLK)#(!CS_REG0&CS_REG1&B  
_CLK)
```

```
 #(CS_REG0&CS_REG1&C_CLK));
```

<프로그램 5. SIO 디코더 PLD LOGIC 프로그램>

"Input

```
A19,A18,A17,A16,IO_0 pin 2,3,4,5,6;
```

"Output

```
OUT pin 19 istype 'com';
```

Equations

```
!OUT = !A19 & !A18 & !A17 & !A16 & !IO_0;
```

<프로그램 6. 채널 선택 디코더 PLD LOGIC 프로그램>

"Input

```
A19,A18,A17,A16,IO_0 pin 2,3,4,5,6';
```

"Output

```
OUT pin 19 istype'com';
```

Equations

```
!OUT = !A19 & !A18 & !A17 & A16 & !IO_0;
```

<프로그램 7. 카운터 PLD LOGIC 프로그램>

```
CNODE1 = DO & D1;
```

```
CNODE2 = CNODE1 & D2;
```

```
CNODE3 = CNODE2 & D3;
```

```
CNODE21 = CNODE20 & D21;
```

```

CNODE22 = CNODE21 & D22;
DO := STOP $ D0;
D1 := D0 $ D1;
D2 := CNODE1 $ D2;
D3 := CNODE2 $ D3;

D21 := CNODE20 $ D21;
D22 := CNODE21 $ D22;
D23 := CNODE22 $ D23;

```

<프로그램 8. 모듈 selection PLD LOGIC 프로그램>

"Input

```
A19,A18,A17,A16,IO_0,IO_1,IO_2 pin 2,3,4,5,6,7,8;
```

"Output

```
OUT pin 12 istype 'com';
```

Equations

```
!OUT = !A19 & !A18 & !A17 & !A16 & !IO_0 & IO_1 & IO_2;
```

end U4