

TDX-1A 시스템 시동

○김관중, 이병선, 김영시
한국전자통신연구소

TDX-1A System Start-Up

Kim Gwan Joong, Lee Byung Sun, Kim Young Si

Electronics and Telecommunications Research Institute(ETRI)

Abstract

The system start-up function of TDX-1A that should be executed during installation or emergency has been designed to meet requirements of the distributed real time control system. It is performed in a centralized manner by the dedicated processor according to the maintenance and administration mechanism of the system. Upon execution, it provides information on the system state for the ESS personal. In additions, it protects the loss the charging and statistics data, and minimizes the effect on the customer service.

This paper describes the design concept of the system start-up function and its implementation method and also calculates its the execution time.

I. 서론

과거의 교환기가 가입자에 대한 호 처리 서비스 능력의 다양성으로 그 시스템의 성능을 평가했는데 비해 최근의 교환기는 시스템 유지보수 및 운용 관리 능력의 다양성 및 효율성을 날로 중대시 하고 있다. 본고는 이러한 시스템 유지보수 기능에 있어서 핵심 역할을 담당하는 시스템 시동(System Start-Up)에 대해서 개략적으로 기술하고, 또한 그 소요 시간을 산출하여 현재 전국 각지에 설치되어 운용중에 있는 TDX-1A 전자 교환기의 시스템 시동 시간을 예측하는데 있어서 기초 자료로 활용될 수 있도록 하였다.

우선 II 장에서는 시스템 시동 기능을 이해하는데 도움을 주기 위해 포괄적으로 TDX-1A 제어계의 구조에 대해 소개하고, III 장에서 시스템 시동 기능의 설계 개념 및 이에 포함되는 프로세서 시동 기능에 관해 논하며, IV 장에서 시스템 시동 시간의 거의 전부를 차지하는 시스템내 각 프로세서의 로딩>Loading> 소요 시간을 TDX-1A 시스템 generic file의 release 1 을 근거로 산출하였고, 마지막으로 V 장에서 결론을 내렸다.

그리고, 본고에서는 TDX-1A 시스템의 모국인 본체에 국한하여 기술하였으며 자국인 원격장치 시스템(Remote Switching System)의 시동 기능¹⁾은 방법상 본체와 동일하므로 별도로 언급하지 않았다.

II. TDX-1A 제어계 구조

TDX-1A 제어 구조²⁾는 근본적으로 부하 및 기능 분산 개념에 의한 완전분산 제어 구조로 구성되어 이러한 분산된 기능은 크게 3 그룹으로 나누어진 프로세서 그룹에 의해 수행되며 계층 구조상 2 레벨의 구조를 갖는다. 즉 그림 1과 같이 TDX-1A 제어계는 2 개의 프로세서 레벨로 구성되어 상위 레벨인 T-그룹 프로세서(이하 TP라 명명)와 하위 레벨 프로세서인 B/D-그룹 프로세서(이하 BP, DP로 각각 명명)로 구성된다.

T-bus를 공유하는 TP들은 서로 수평적인 관계를 가지며, B/D-bus를 공유하는 BP 및 DP 와는 수직적 관계를 유지한다. 하위 레벨인 BP는 가입자 회로, 트렁크 회로 및 각종 신호 장비를 포함하는 통회로계 장치를 직접 제어하고, DP는 magnetic tape driver, disk driver 및 CRT 등의 유지보수용 시스템 주변 장치

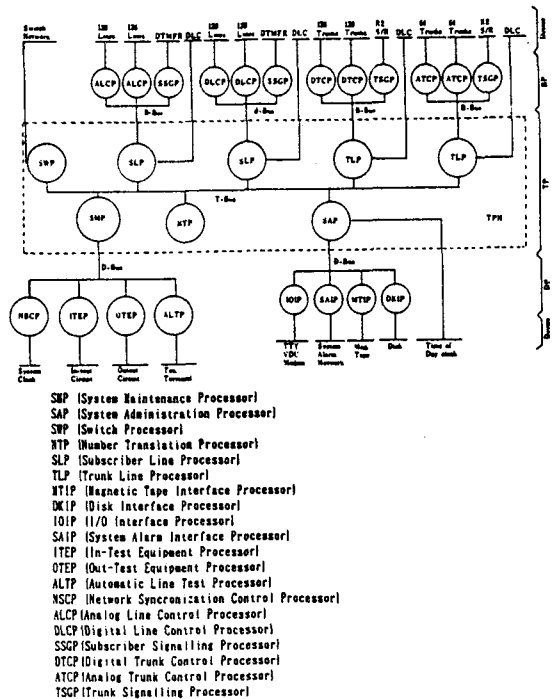


그림 1. TDX-1A 제어계 구조

를 제어하며 경보 기능을 담당한다. 또한 기능적으로 분산된 상위 레벨인 TP는 하위 레벨인 BP/DP로 부터 발생된 각종 event를 받아 각각 전반적인 호처리 기능과 시스템의 유지보수 및 운용 관리 기능을 수행하며 그 결과를 다시 BP/DP로 전송함으로써 전체 교환 기능이 이루어진다. 그리고, 본 시스템의 프로세서 상호 간의 연결 구조는 multi-drop 형태의 global bus이며 프로세서간의 통신은 모든 프로세서가 순차적으로 bus turn을 받아 메시지를 전송하는 round robin 방식이다.

또한, BP와 DP 중 SMP 하위에 속하는 DP는 ROM으로 기능이 실현된 반면에 이중화된 TP와 SAP 하위에 속하는 DP는 O.S.를 제외하고 전 기능이 RAM으로 실현되어 시동시 보조기억 장치(DISK)로 부터 자신의 프로그램 및 데이터를 로딩 받는다.

III. 시스템 시동

시스템의 초기 설치시 정상 서비스가 가능할때까지 수행되는 모든 과정을 시스템 시동이라 하며, 시스템이 정상 서비스 수행중에 정전이나 H/W 및 S/W 장애로 인해 기능이 훼손된 상태에서 시스템이 복구되어 정상 서비스가 가능하도록 수행되는 과정을 시스템 재시동이라 하는데, TDX-1A 시스템에서의 기능 구현시 재시동은 시동의 범주에 포함이 되도록 하였으므로 시스템 시동에 관해서만 기술하기로 한다.

1. 설계 개념

1) 자동적인 로딩

TP 및 SAP 하위의 DP는 시동시 ROM에 실장된 bootstrap loader에 의해 자동적으로 로딩을 받는다.

2) 절대주소에 의한 load segment allocation

최근 대용량의 전자 교환기는 충분한 메모리 용량을 바탕으로 일부 개발 도구들이 시스템에 내장되어 있으며 generic file을 해당 프로세서로 로딩시 relocatable allocation이 가능하다. 반면에 TDX-1A 시스템은 중소 용량의 전자 교환기로서 시스템의 generic file은 원시코드가 Z-80 어셈블리 언어인 목적코드를 work station 상에서 사전에 link하여 직접 수행할 수 있는 형태의 procedure file이므로 이를 해당 프로세서로 로딩시에 각 load segment를 지정된 절대주소의 RAM 영역에 할당하게 된다.

3) 전담 프로세서에 의한 로딩 수행

시스템 시동 기능과 같이 시스템의 구조와 밀접한 관계를 갖는 기능은 가능 설계시 우선 그 구조적 특징을 반영해야 하고, 또한 시스템내에서 각 프로세서가 담당하는 역할을 고려해야 한다. 본 시스템은 그림 1에서 보는 바와 같이 모든 TP가 로딩용 보조기억 장치인 DISK를 제어하는 프로세서인 DKIP와 통신하기 위해서는 반드시 SAP를 통해야 하고, 또한 시스템 유지보수 기능은 그 특성상 SMP에 의해 중앙집중 방식에 의해 총괄적으로 수행되며¹⁾ DKIP를 다른 DP와 같이 단지 device controller로 보는 견지에서 본 시스템의 시동 기능은 SAP와 SMP에 의해 주도적으로 수행되고, DKIP는 DISK의 generic file을 access하는 기능만 수행하도록 설계하였다. 그리고, 나머지 TP와 SAP 하위의 DP는 자신의 프로그램 및 데

이타를 SAP를 통하여 DKIP로 부터 받아 지정된 RAM 영역에 할당하는 기능만이 모든 프로세서에 공통적으로 실장된다.

4) 프로세서별 로딩 우선순위 적용

시스템 재시동을 시동 절차의 일부로 포함시킬수 있도록하여 시동시에도 프로세서 기능에 따른 로딩 우선순위를 적용한다. SAP는 모든 프로세서에 우선하여 로딩을 받아야 하고, 다음 시스템 시동 상황을 운영자가 파악할 수 있도록 입출력 담당 프로세서인 IOIP가 로딩을 받아야 한다. 그리고, TP보다 DP가 먼저 로딩을 받아 서비스 환경 조성을 우선적으로 준비하며, SAP와 더불어 로딩 담당 프로세서인 SMP는 다른 TP보다 먼저 로딩을 받아야 하므로 즉, DKIP, SAP, IOIP, 나머지 DP, SMP, 나머지 TP 순의 우선순위를 설정한다.

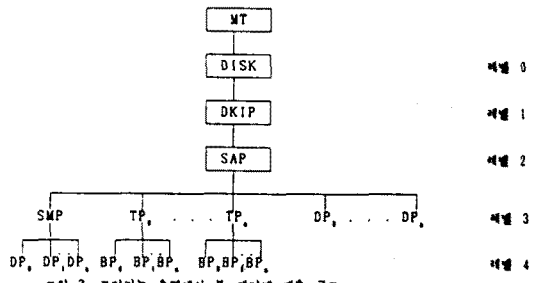
5) Bank 단위 로딩

SAP는 각 프로세서에 대한 로딩 수행시 프로세서별 physical memory size인 bank(64 Kbyte) 단위로 운용자에게 로딩 진행 상황(시작, 완료, 실패)을 보고하고, 로딩 우선순위를 적용하며 또한 bank 단위로 로딩 size 정보를 관리한다.

2. 시스템 시동 순서

상기 설계 개념에 따라 시스템 시동 기능을 실현하면 시스템 시동시 시스템내 각 프로세서는 다음과 같은 순서로 시동하여 프로세서 각각의 고유 기능을 수행한다. 그림 2는 그림 1의 TDX-1A 계층 구조를 DKIP 입장에서 재 구성한 것으로써 시스템은 5 레벨의 계층 구조를 갖는데 이는 다음과 같은 특징이 있다.

- 상위 레벨의 프로세서가 로딩을 받아 시동하지 않으면 하위 레벨에 속하는 프로세서는 로딩을 받을수 없다.
- ROM에 기능이 실장된 레벨 4의 프로세서들은 레벨 3의 차 상위 프로세서가 정상 서비스를 개시하는 시점에서 시동한다.
- 레벨 3의 프로세서들은 앞서 기술된 프로세서별 우선순위에 따라 순차적으로 로딩을 받는다.



1) MT에서 DISK로의 초기 로딩

시스템 설치시 또는 소프트웨어 교체시에 최초로 수행되며 시스템 재시동시에는 수행되지 않는다.

2) DKIP 시동

시스템 generic file이 실장된 DISK를 제어하는 프로세서인 DKIP가 시스템 시동시 가장 먼저 스스로 DISK로 부터 자신의 프로그램 및 데이터를 로딩받아 가장 먼저 정상 서비스 상태로 천이하여 다른 프로세서에 대한 로딩 서비스를 수행한다.

3) SAP 시동

DKIP를 관장하며 전 프로세서에 대한 로딩 수행을 담당하는 프로세서이므로 다른 프로세서에 우선적으로 로딩을 받아 시동한다.

4) SAP 하위의 DP 시동

입출력 담당 프로세서 IOIP가 로딩을 받아 시동하여 운용자가 시스템의 동작 상태를 파악할 수 있도록 하며 나머지 DP는 FIFS(First-In-First-Service)의 원칙에 따라 수행된다.

5) SMP 시동

SMP는 TP에 대한 로딩 수행을 주관하고 유지보수 수행 체계상 SMP가 모든 유지보수 기능을 총괄하므로 타 TP에 비하여 우선적으로 로딩을 받아 시동한다.

6) 나머지 TP 시동

SMP를 제외한 나머지 TP가 FIFS 원칙에 따라 로딩을 받아 시동한다.

7) BP 및 SMP 하위의 DP 시동

Generic 프로그램이 ROM화된 프로세서들로서 차 상위 TP가 정상 상태로 천이하는 시점에서 상위 TP의 명령에 의해 정상 서비스를 개시한다.

3. 프로세서 시동 과정

시스템 시동시 시스템내 각 프로세서는 자신이 속한 그룹(TP, BP, DP)에 따라 하드웨어 초기화 및 시험을 수행하고, 자신의 프로그램과 데이터를 로딩 받으며 서비스를 시작하기 위하여 자신의 소프트웨어를 초기화시키는 3 단계의 프로세서 시동 과정을 수행한후 정상 서비스 상태로 천이한다. 모든 프로그램이 ROM화 되어있는 BP 및 DP들은 로딩 단계를 수행하지 않는다.

1) 하드웨어 초기화 및 시험

TDX-1A 시스템은 각 프로세서의 CPU로 Z-80을 사용하므로 프로세서 시스템의 각종 제어용 주변 장치 또는 정합 장치도 Z-80 계열의 장치이다. 이러한 H/W 적인 특징으로 인하여 모든 프로세서는 자신의 power off/on, H/W reset 단자의 단락, 프로세서내에서 긴급 장애 발생시 Z-80 CPU와 각 장치에 고유한 초기화가 수행된다. 또한 프로세서의 시동의 반복 또는 시동 이후의 malfunction 발생등을 방지하기 위하여 하드웨어 기본 시험¹⁶⁾(CPU, 메모리, Bus 시험등)을 수행하며 로딩을 받기위한 bus 통신 관련 환경 조성도 병행된다.

2) 로딩 단계

상기 단계를 수행한 TP 및 SAP 하위의 DP 프로세서들은 다음과 같은 절차에 따라 DISK로부터 로딩을 받는데 그런 3은 정상적인 경우만을 고려하여 이를 도식화한 것이다.

- (1) TP 및 SAP 하위의 DP는 SMP/SAP로 로딩을 각각 요청한다.
- (2) TP에서 로딩 요청을 받아 SAP로 이 사실을 통보한다.
- (3) 해당 프로세서의 첫번째 load segment를 요청한다.
- (4) 로딩 시작한 사실을 운용자에 알린다.
- (5) 첫번째 load segment(length : 256 byte)를 받는다.
- (6) Load segment를 해당 프로세서로 보낸다.

- (7) Handshake signal(Acknowledge)을 받는다.
- (8) 다음 load segment를 요청한다.
- (9) 해당 프로세서의 마지막 load segment를 요청한다.
- (10) 해당 segment를 받는다.
- (11) Segment를 해당 프로세서로 보낸다.
- (12) Handshake signal(Acknowledge)을 받는다.
- (13) 특정 프로세서에 대한 로딩이 완료되었음을 알린다.
- (14) SMP로 이 사실을 보고한다.
- (15) 운용자에 이 사실을 알린다.

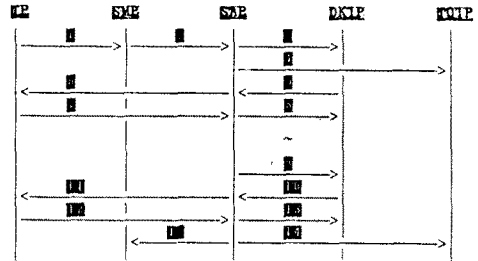


그림 3. 프로세서 로딩 수행 절차

3) 소프트웨어 초기화 단계

각 프로세서가 상기의 과정을 수행후 정상 서비스 상태로 천이하기 이전에 이를 준비하는 단계로써 정상 상태에서 수행될 각종 기능 수행을 준비하고 이에 따른 데이터를 초기화한다.

IV. 시스템 시동 시간 계산

1. 조건

- 1) 프로세서 자체 기능 및 프로세서간의 통신 기능은 정상으로 가정하며 로딩 기능 수행도 외부 영향에 의한 중간 어려움이 정상 수행하는 것으로 가정한다.
- 2) 로딩 관련 메시지 또는 이의 서비스량에 따라 달라질 수 있는 소요 시간의 경우는 평균 시간으로 계산한다.
- 3) TP 및 SAP 하위 DP에 실장되는 O.S 및 로딩 기능은 동일한 것으로 간주한다.
- 4) Z-80 CPU의 cycle time은 2.5 MHz 이다.

2. 주요 인자

1) t_1

프로세서가 메시지를 출력하려면 우선 해당 메시지를 output queue에 write하고 전송을 요구한후 bus turn이 돌아오면 그때 전송한다.¹⁷⁾ 그런데 해당 프로세서는 RTC(Real Time Clock, 주기 : 50 ms) 단위로 메시지 전송을 요구하므로 이때까지 걸리는 시간은 평균적으로 25 ms로 볼 수 있고, 해당 프로세서에 bus turn이 돌아올때까지의 대기 시간은 bus turn around time¹⁸⁾을 고려하면 20 ms가 된다. 즉, 메시지 출력시 총 대기시간은 평균적으로 45 ms로 계산할 수 있다.

2) t_2

Input queue에 메시지가 입력된후 부터 그 메시지가 O.S.에 의해 처리될때 까지 걸리는 시간에 대한 평균치는

입력된 메시지 처리 주기(1 RTC, 50 ms)의 1/2, 즉 25 ms로 계산할 수 있다.

- 3) t_b
- Bus 상에서 데이터의 전송 속도는 256 Kbps 이다. 따라서 메시지 length(L)에 따른 전송 시간은 다음과 같다. (단, 1L 은 12 byte 이다.)
- L 이 1 인 경우 $t_{b1} = 0.37 \text{ ms}$
 - L 이 23 인 경우 $t_{b2} = 8.42 \text{ ms}$

- 4) t_s
- 해당 로딩 기능에서의 수행시간.
- SMP의 로딩 요청(load request) 메시지 처리시간(t_{s1}) : 0.88 ms
 - SAP의 로딩 수행 명령(load command) 메시지 처리시간(t_{s2}) : 2.98 ms
 - SAP가 DKIP로 부터 프로그램 및 데이터(program/data)를 받아 처리하는 시간(t_{s3}) : 5.91 ms
 - 로딩 받는 프로세서가 자신의 프로그램 및 데이터를 RAM 영역에 allocate하는 시간(t_{s4})
 - Base(bank0) memory 영역 로딩 (t_{s5}) : 3.03 ms
 - Ext.(bank1,2,3) memory 영역 로딩 (t_{s6}) : 3.47 ms
 - SAP가 로딩 받는 프로세서로 부터 handshake signal을 받아 DKIP에 다음 load segment를 요청하는 시간(t_{s7}) : 1.53 ms

- 5) l
- 로딩 받는 RAM 영역의 size를 나타내며 각 프로세서별 이에 관한 정보는 표 1과 같다.
- 단, l_1 : Base(bank0) memory 영역의 size
 l_2 : Ext.(bank1,2,3) memory 영역의 size

표 1. 프로세서별 로딩 size(K byte)

Proc. name \ Mem.	Bank0	Bank1	Bank2	Bank3	Total
SMP	40	64	16	-	120
SAP	40	60	-	-	100
SWP	40	4	4	4	52
NTP	40	64	64	52	220
SLP	40	64	20	-	124
TLP	40	64	4	-	108
MTIP	36	4	-	-	40
IOIP	36	4	64	-	104
SAIP	36	4	-	-	40
DKIP	36	4	-	-	40

- 6) t_c
- DKIP가 SAP로 부터 load segment 요청을 받아 이를 서비스하는데 걸리는 시간이며 다음 두가지 요소의 합으로 결정된다.
- DKIP가 DISK에서 해당 프로세서의 file을 read 하는데 걸리는 시간 : 2.09 ms
 - File read 시간을 제외한 순수한 load segment 메시지 처리 시간 : 3.52 ms
- 즉, 총 DKIP의 서비스 시간은 5.61 ms가 된다.

3. 프로세서 로딩 시간 계산

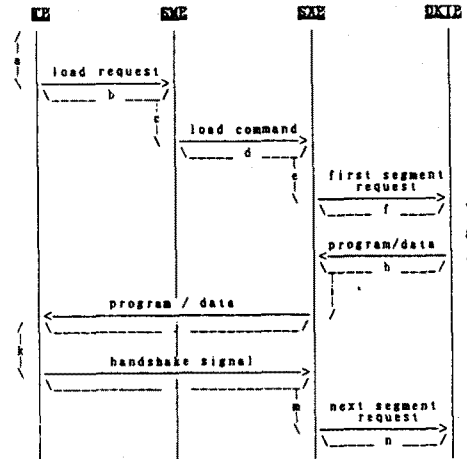


그림 4. TP의 로딩 절차

상기 그림 4의 각 단계에서의 처리시간 $T_a, T_b, T_c, T_d, T_e, T_f, T_g, T_h, T_i, T_j, T_{k1}, T_{k2}, T_l, T_m, T_n$ 은 다음과 같다.

$$\begin{aligned}
 T_a &= t_s = 45 \text{ ms} \\
 T_b &= T_d = T_f = T_l = T_n = t_{b1} = 0.37 \text{ ms} \\
 T_c &= t_1 + t_{s1} + t_s = 70.88 \text{ ms} \\
 T_e &= t_1 + t_{s2} + t_s = 72.98 \text{ ms} \\
 T_g &= t_1 + t_s + t_c = 75.61 \text{ ms} \\
 T_h &= T_j = t_{b2} = 8.42 \text{ ms} \\
 T_i &= t_1 + t_{s3} + t_s = 75.91 \text{ ms} \\
 T_{k1} &= t_1 + t_{s4} + t_s = 73.03 \text{ ms} \\
 T_{k2} &= t_1 + t_{s5} + t_s = 73.47 \text{ ms} \\
 T_m &= t_1 + t_{s6} + t_s = 71.53 \text{ ms}
 \end{aligned}$$

즉, 각 TP의 로딩 시간

$$\begin{aligned}
 T &= T_a + T_b + T_c + T_d + T_e + T_f \\
 &+ 4l_1(T_g + T_h + T_i + T_j + T_{k1} + T_l + T_m + T_n) \\
 &+ 4l_2(T_g + T_h + T_i + T_j + T_{k2} + T_l + T_m + T_n) \\
 &- (T_g + T_m + T_n)
 \end{aligned}$$

또한 DP는 로딩 요청을 SAP로 하고, SAP는 직접 DKIP와 고신하여 로딩을 받는 것을 제외하고는 모든 로딩 수행 절차가 TP와 동일하므로 이에 대한 구체적인 로딩 소요 시간의 산출은 생략한다. 상기 수식을 근거로 계산된 시스템내 각 프로세서에 대한 로딩 소요 시간은 표 2와 같다.

표 2. 프로세서별 로딩 소요시간(초)

Proc. Name	소요 시간
SMP	159.74
SAP	63.05
SWP	65.31
NTP	276.38
SLP	155.77
TLP	135.66
MTIP	50.24
IOIP	130.65
SAIP	50.24
DKIP	3

표 3. 프로세서별 최대 실행 개수

Proc. Name	개 수
SMP	1
SAP	1
SWP	1
NTP	1
SLP	20
TLP	8
MTIP	3
IOIP	3
SAIP	1
DKIP	2

4. 시스템 시동 시간

상기 II의 2에서 기술한 바와 같이 시스템 시동시 시스템 내 각 프로세서는 DKIP로 부터 시작하여 RAM으로 기능이 실현된 모든 프로세서들이 로딩을 받고 마지막으로 BP가 시동함으로써 시스템은 정상 서비스를 시작한다. 즉, 시스템 시동 시간은 총 로딩 소요 시간에 BP가 시동하는데 걸리는 시간을 더하면 된다. 표 3은 본 시스템의 full configuration시의 프로세서별 실장 갯수를 나타내는데 BP 시동 시간 5초를 고려하면 이때의 총 소요시간은 1시간 29분 14초이다. 그리고, 시스템 운용중 일부 프로세서의 재시동 시간은 프로세서별 로딩 시간으로 산출할 수 있다.

V. 결론

한 시스템의 시동 기능은 그 시스템의 구조, 기능의 특성과 밀접한 관계가 있으며 초기 규격 작성 단계에서 시스템의 개발 비용, 신뢰도 및 서비스의 질에 대한 충분한 검토가 이루어져야 한다.

최대 32개의 상위 레벨 프로세서와 200여 개의 하위 레벨 프로세서로 분산제어 방식에 의해 구현된 TDX-1A 시스템은 그 규모와 기능의 복잡성에 의해 단순한 타 시스템의 시동 방법은 사용할 수 없으며, 프로세서와 보조기억 장치(DISK)의 이중화를 이용하여 시동 기능의 신뢰도를 최대한 보장하고 기능구현 알고리즘의 최적화로서 서비스 중지 시간을 최소화할 수 있도록 노력하였다.

현재 TDX-1A 시스템은 전국 각지에 설치되어 운용중에 있으며 시스템 시동 기능은 시스템 설치시 혹은 긴급 상태 발생시의 복구시에 정상적으로 수행되고 있다.

본고에서 제시한 TDX-1A 시스템 시동 기능의 설계개념, 구현 방법은 현재 개발이 진행중인 TDX-10 시스템의 개발과 이와 유사한 다른 대형 시스템 개발에 유용하게 활용될 수 있을 것으로 기대된다.

감사의 말씀

TDX-1A 전전자 교환기 개발 사업중 시스템 시동 기능 개발을 주도하신 천유식 부장님, 유환영 부장님께 감사드리고, 아울러 본 기능의 개발 초기에 참여했던 황종범, 주성순씨와 본 기능 개발에 많은 협력을 해주신 김태준, 김철수, 이호진씨께도 감사드립니다.

참고 문헌

- [1] 김관중, "긴급 복구", 한국전자통신공사 연수원 교재, 목록번호 36-51-21, 1986.9
- [2] 이현, 송광석, 임태원, "제어계 구조", 전자통신, Vol.8, No.2, Jul.1986
- [3] 이병선, 김영시, 이윤복, "TDX-1 유지보수 기능", 전자통신, Vol.8, No.2, Jul.1986
- [4] 한국전자통신연구소, "TDX-1 MA", TDX-1 기술전수교재 제 4권, 1985.1.
- [5] 김관중, 이재섭, "TDX-1 교환기에 있어서의 프로세서계 유지보수", 대한전자공학회 통신교환 합동학술 발표회 논문집, Vol.10, No.1, 1986.11.
- [6] 김관중, "프로세서 유지보수", 한국전자통신공사 연수원 교재, 목록번호 36-51-11, 1986.9
- [7] 한국전자통신연구소, "TDX-1 Operating System, TDX-1 기술전수교재 제7권, 1985.1.
- [8] 한국전자통신연구소, "전전자교환기 개발 사업중 제어계개발", 연구과제 보고서, 1983.12