

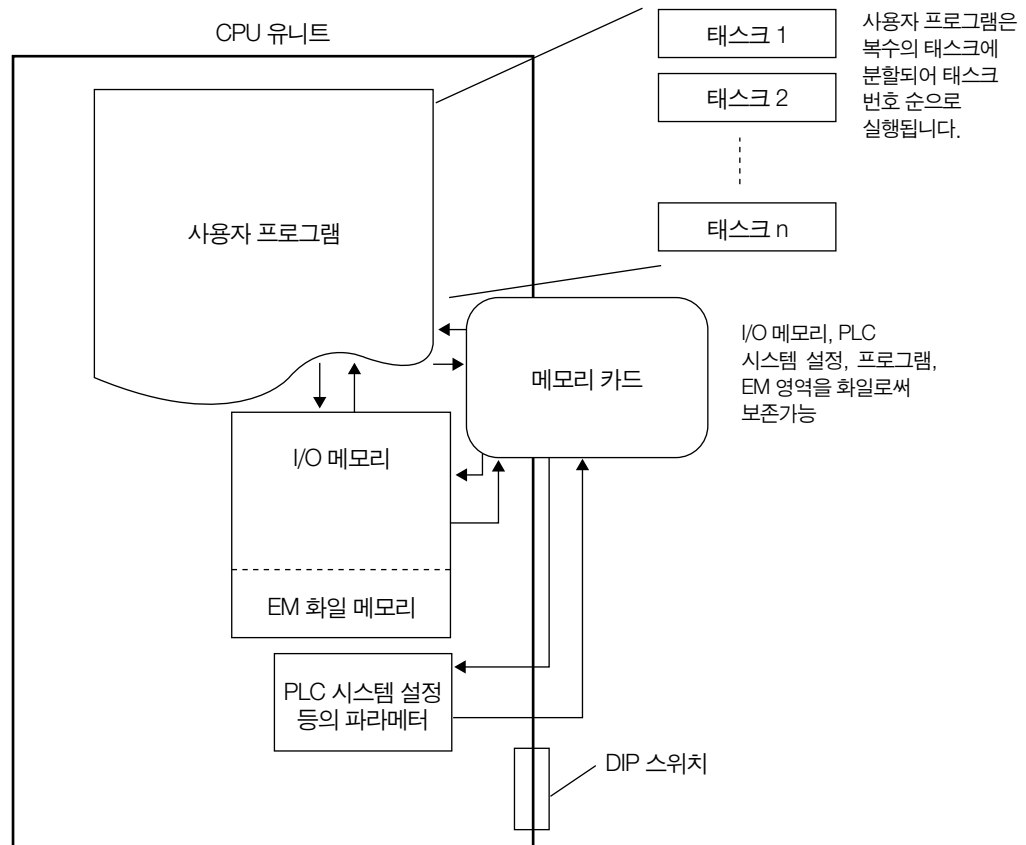
## 제 6 장

# CPU 유닛 동작의 개요

## 6-1 CPU 유닛 내부의 구조

### ■ CPU 유닛 내부의 구조

CPU 유닛 내부의 구조는 다음과 같이 되어 있습니다.



### 사용자 프로그램

사용자 프로그램은 복수의 프로그램으로부터 성립되고 프로그램은 인터럽트도 포함하여 최대 288개 작성 가능합니다. 주변 툴 CX-Programmer에 의해 이들 프로그램은 실행 단위인 태스크에 1:1로 할당된 후 CPU 유닛에 전송됩니다.

태스크에는 1사이클에 1회 실행되는 태스크 (사이클 실행 태스크, 최대 32개)와 인터럽트 조건이 성립한 때만 실행되는 태스크 (인터럽트 태스크, 최대 256개)가 존재하며, 사이클 실행 태스크는 번호순으로 실행됩니다.

프로그램에 기술된 명령은 프로그램 선두에 어떤 것으로부터 차례로 실행되어 I/O 메모리를 읽고 씁니다. 모든 사이클 실행 태스크를 실행하면 각 유닛과의 I/O 리프레쉬를 하고 가장 번호가 작은 사이클 실행 태스크부터 다시 실행을 반복합니다 (사이클릭 스캔).

## I/O 메모리

사용자 프로그램이 읽고 쓰기 하는 RAM 영역입니다. 전원단 복귀시에 클리어 되는 영역과 유지하는 영역으로부터 성립됩니다.

또 각 유닛과 데이터 교환되는 영역과 내부에서 사용하는 영역으로 나뉩니다.

각 유닛과의 데이터 교환은 명령 실행 사이클에 1회와 명령실행시에 즉시의 2가지 방법이 가능합니다.

## PLC 시스템 설정

소프트 스위치에 의해 각종 초기설정 등을 행합니다.

## DIP 스위치

하드 스위치에 의해 초기설정 등을 행합니다.

## 메모리 카드

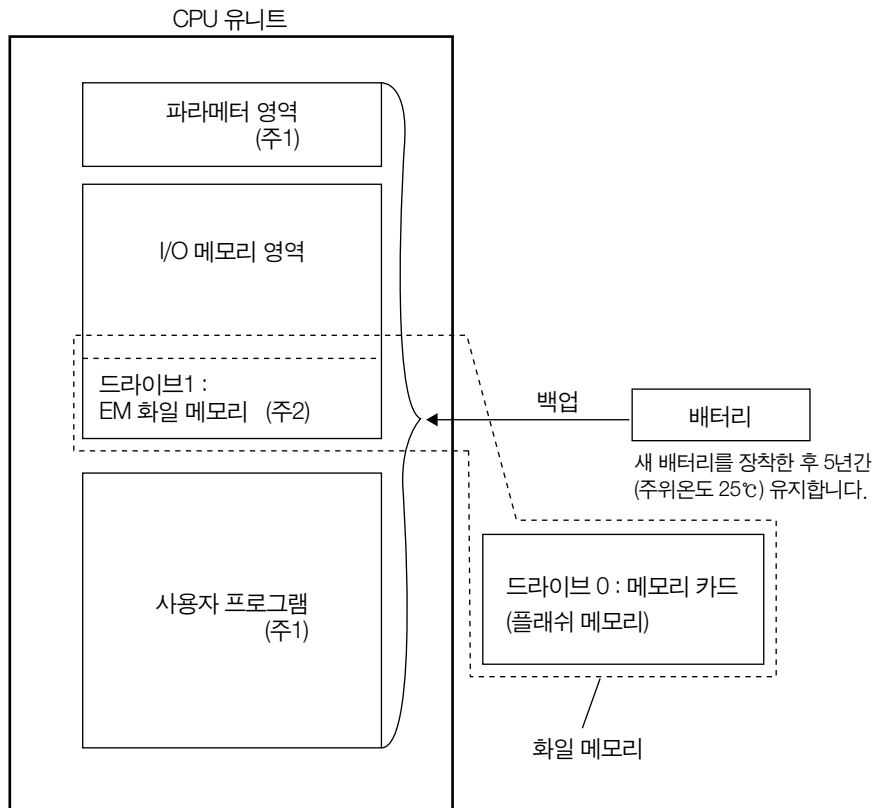
필요에 따라 사용합니다. 프로그램, I/O 메모리, PLC 시스템 설정, 주변 톨로 작성한 I/O 코멘트 등을 보존할 수 있습니다. 전원 ON시에 메모리 카드로부터 프로그램과 각종 시스템 설정을 자동 읽고 쓰기도 가능합니다 (전원 ON시 자동 전송 기능).

## ■ CPU 유닛의 메모리 블록도

CS1 시리즈의 CPU 유닛 메모리(RAM)은 아래의 블록 구성으로 되어 있습니다.

- 파라미터 영역 (PLC 시스템 설정, 등록 I/O 테이블, 루틴 테이블, CS1 CPU 고기능 유닛 시스템 설정 영역)
- I/O 메모리 영역
- 사용자 프로그램 (태스크 정보를 포함)

상기 영역은 배터리 (CS1W - BAT01)에 의해 백업되어 있습니다. 배터리 전압이 저하하면 이들 영역의 데이터는 소실됩니다.



주1 : CPU 유닛 전면의 DIP 스위치 SW1을 ON으로 하는 것에 의해 쓰기 불가로 하는 것이 가능한 영역은 파라미터 영역과 사용자 프로그램입니다.

주2 : EM 파일 메모리는 확장 DM(EM) 영역의 지정 बैं크 이후를 파일 메모리화 해서 데이터 파일과 프로그램 파일을 격납하는 것이 가능한 RAM 메모리 모체 (메모리 카드와 동일 형태인 위치의 모체)입니다.

### 주 의

CPU 유닛을 처음 사용할 때 반드시 부속 배터리 (CS1W-BAT01)를 장착하십시오. 그 후, 주변 툴을 사용하여 메모리 올 클리어 조작을 행하는 것으로 상기 메모리 (RAM) 내의 파라미터 영역, I/O 메모리 영역, 사용자 프로그램을 클리어 하십시오.

## 6-2 동작 모드

### ■ 동작 모드의 개요

CPU 유닛에는 다음 3가지의 동작 모드가 있습니다. 동작 모드는 사용자 프로그램 전체를 제어하고, 전 태스크에 공통입니다.

#### 「프로그램」 모드

프로그램의 정지 상태, 다음과 같은 프로그램 실행전의 준비를 이 모드에서 행합니다.

- I/O 테이블 작성
- PLC 시스템 설정 등의 초기 설정
- 프로그램의 전송 체크
- 강제 세트/리세트에 의한 배선 체크 등

이 모드에는 모든 사이클 실행 태스크는 미실행 상태 (INI) = 정지 상태입니다.

또, 모든 인터럽트 태스크는 정지 상태입니다. 자세한 것은 6-4항 「태스크의 개요」를 참조하십시오.

역시, 이 모드에서도 I/O 리프레쉬는 행해집니다.

### 경고

운전을 정지하고 있는 상태 (「프로그램」) 모드에 있어서도 CPU 유닛은 I/O 리프레쉬를 행하고 있습니다. 따라서 아래의 어떤 조작에 의해 출력 유닛에 할당된 출력 릴레이 영역의 데이터, 또는 고기능 I/O 유닛/CPU 고기능 유닛에 할당된 각 릴레이 영역의 데이터를 변경하는 경우, 충분히 안전을 확인한 후 행하십시오. 고기능 I/O 유닛 / CPU 고기능 유닛에 접속시킨 부하가 생각하지 않던 동작을 할 위험이 있습니다.

- 주변 툴 (PC 툴)에 의해 I/O 메모리의 CPU 유닛으로 전송 조작
- 주변 툴에 현재치 변경조작
- 주변 툴에 의해 강제 세트/리세트 조작
- 메모리 카드 또는 EM 화일 메모리로부터의 I/O 메모리 화일의 CPU 유닛으로의 전송 조작
- 네트워크 상의 다른 PLC 또는 상위 컴퓨터로부터의 I/O 메모리의 전송 조작



### 「모니터」 모드

프로그램의 실행상태에서 주변 톨에 의함, 다음과 같은 각종 설정 조작이 가능합니다.  
시운전 조정 등을 이 모드에서 행합니다.

- 온라인 에디트
- 강제 세트/리세트
- I/O 메모리의 현재치 변경 등

이 모드에는 「운전 개시시 기동」 속성을 갖는 싸이클 실행 태스크 또는 태스크 기동 명령 (TKON)에 의해 기동된 싸이클 실행 태스크는 실행 가능 상태 (READY)로 되고, 실제로 실행권이 돌아서 오면 실행됩니다. 인터럽트 태스크는 실행 조건이 성립하면 실행됩니다.

### 「운전」 모드

프로그램의 실행 상태, 주변 톨에 의함, 온라인 에디트, 강제 세트/리세트, I/O 메모리의 현재치 변경 등의 각종 설정 조작은 불가능합니다. 주변 톨에 의한 프로그램 실행 상황의 모니터 (프로그램 모니터, I/O 메모리 모니터)는 가능합니다. 시스템의 본운전을 이 모드에서 행합니다. 이 모드에는 「운전개시시 기동」 속성을 갖는 싸이클 실행 태스크, 또는 태스크 기동 명령 (TKON)에 의해 기동된 싸이클 실행 태스크는 실행 가능 상태(READY)로 되고, 실제로 실행권이 돌아서오면 실행됩니다. 인터럽트 태스크는 실행조건이 성립하면 실행됩니다. 각 동작 모드에 있어서 가능한 설정 조작의 자세한 것은 15-2항 「CPU 유닛의 동작모드」를 참조하십시오.

## ■ 동작 모드의 변경과 I/O 메모리의 관계

| 모드변경                   | 비유지 영역 (주1) | 유지영역 (주2) |
|------------------------|-------------|-----------|
| 「운전」 또는 「모니터」 → 「프로그램」 | 클리어 (주3)    | 유지        |
| 「프로그램」 → 「운전」 또는 「모니터」 | 클리어 (주3)    | 유지        |
| 「운전」 ↔ 「모니터」           | 유지          | 유지        |

주1 : 비유지 영역 : CIO, 내부 보조 릴레이 - WR, 내부 보조 릴레이, 타이머 현재치/플래그, 인덱스 레지스터, 데이터 레지스터, 태스크 플래그, 컨디션 코드  
(특수 보조 릴레이는 어드레스에 따라 틀립니다.)

주2 : 유지영역 : 유지 릴레이, 데이터 메모리, 확장 데이터 메모리, 카운터 현재치/플래그

주3 : I/O 메모리 유지 플래그를 1(ON)로 설정하고 있는 경우는 유지됩니다. 단, 운전정지 이상 (FAL 명령 실행을 포함)이 발생해서 운전이 정지한 때는 CPU 유닛내의 I/O 메모리는 유지됩니다. 만, 출력 유닛의 출력 접점은 전부 OFF로 됩니다.

I/O 메모리의 자세한 것에 대해서는 제7장 「I/O 메모리 등의 상세」를 참조하십시오.

## ■ 전원 ON시의 동작 모드

전원 ON시의 CPU 유닛의 동작 모드에 대해서는 8-5항 「PLC 시스템 설정 내용의 설명」을 참조하십시오.

### 주 의

전원 ON시 동작모드를 「모니터」 모드 또는 「운전」 모드로 하기 위해서는 PLC 시스템 설정의 내용을 초기값에서 변경하십시오.

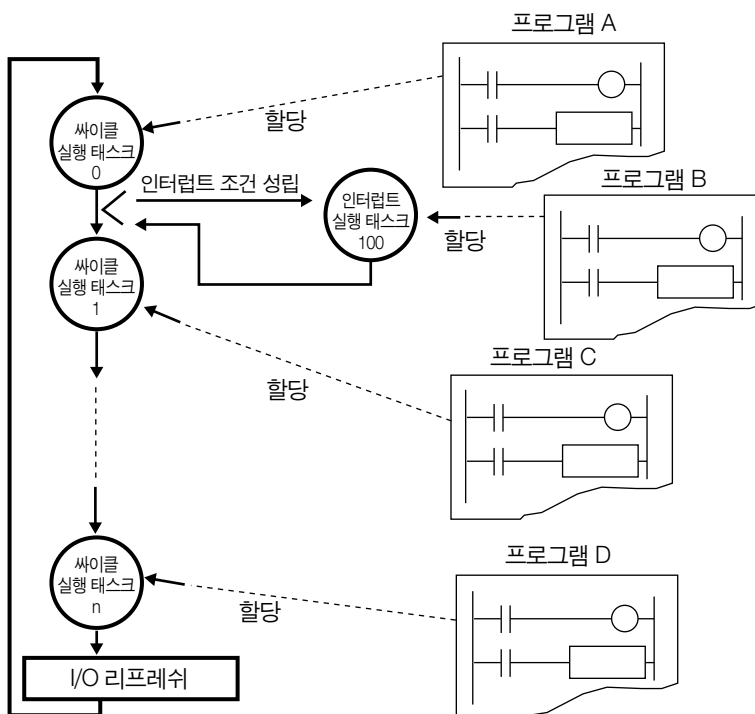
## 6-3 프로그램과 태스크

태스크란 각각의 프로그램을 어떤 순서 또는 인터럽트 조건으로 실행시키는지 지정하는 기능입니다. 태스크에는 크게 나뉘어 아래의 2종류가 있습니다.

- 1) 순서대로 실행되는 태스크를 「싸이클 실행 태스크」라고 부릅니다.
- 2) 인터럽트 조건대로 실행되는 태스크를 「인터럽트 태스크」라고 부릅니다.

「싸이클 실행 태스크」에 할당된 프로그램은 태스크 번호순으로 실행됩니다.

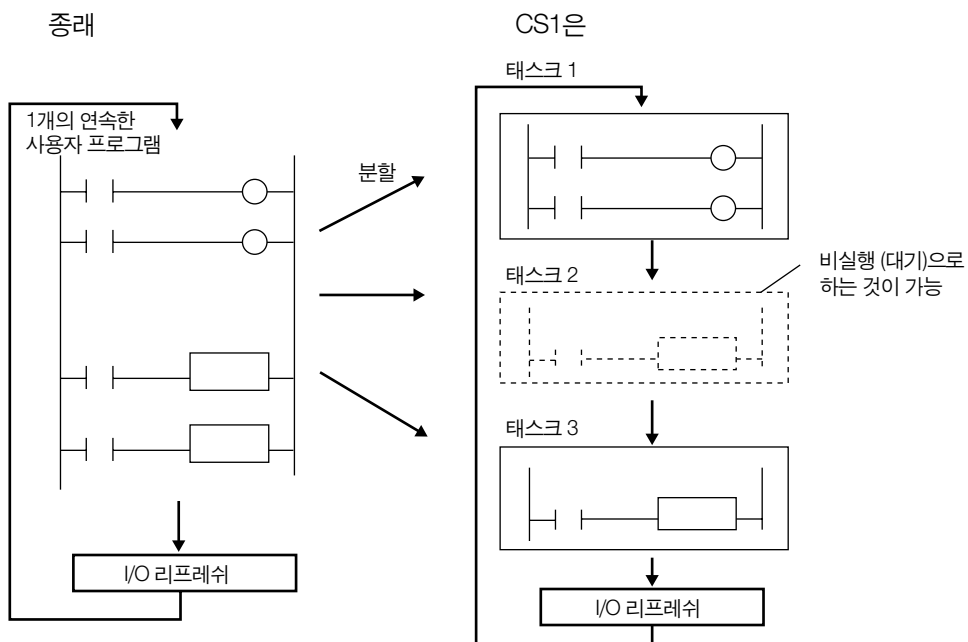
전 태스크 (정확히는 실행가능 상태의 태스크)를 실행한 후 I/O 리프레쉬가 1싸이클에 1회 실행됩니다. 그 사이에 인터럽트 조건이 성립한 경우 그 인터럽트 태스크에 할당시킨 프로그램이 실행됩니다.



종래의 PLC에서 본 관점에서는 1개의 연속한 프로그램을 얼마가를 분할한 형태로 됩니다. 단, 각 태스크에 할당한 프로그램은 종래와 같이 최후에 END 명령을 갖는 1개의 프로그램입니다.

태스크의 특징은 그 실행 및 비실행 (「대기」라고 부름)을 각 사이클 실행 태스크 간에 「태스크 제어 명령」에 따라 임의로 제어할 수 있다는 점입니다.

따라서 얼마가의 프로그램 부품을 태스크로써 조합해 두어 품종과 공정 등에 따라 어떤 프로그램(태스크)만을 실행하는 것이 가능합니다. (프로그램의 단계 전환) 또, 필요한 때에 필요한 프로그램만을 실행 시키는 것이 가능하기 때문에 성능 (사이클 타임) 향상에도 공헌합니다.



역시, 일단 실행시킨 태스크는 다음의 사이클 이후에도 실행됩니다. 일단 대기시킨 태스크는 다른 태스크부터 실행으로 돌리지 않는 한 다음 사이클 이후에도 대기시킨 그대로입니다.

#### 참 고

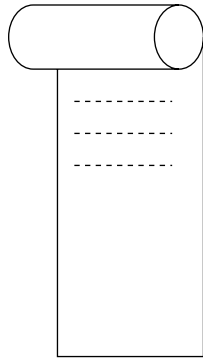
말하자면, 종래의 프로그램이 처음부터 “두루말이”를 읽어가는 이미지인 것에 대해, 태스크는 각 카드를 각각 읽어가는 이미지입니다.

단,

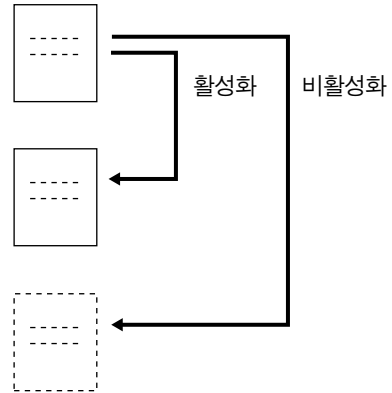
- 각 카드의 읽는 순번은 결정되어 있어 순서대로 읽어 갑니다.
- 각 카드는 상호 활성 / 비활성을 지시할 수 있고, 비활성시킨 카드는 읽어 날립니다 (활성 / 비활성의 지시는 태스크 제어 명령으로 행합니다).
- 일단 활성화된 카드는 다음의 읽는 순에도 활성 그대로 읽어집니다.

일단 비활성된 카드는 다른 카드로부터 복활 (활성화) 시키지 않는한 비활성 그대로 읽어 날립니다.

종래의 프로그램 : 두루말이



CS1 : 카드, 더구나 상호 활성화 · 비활성을 지시



## 6-4 태스크의 개요

태스크에는 크게 나눠 다음의 2종류가 있습니다.

### 1. 싸이클 실행 태스크 (최대 32개)

1싸이클에 1회, 태스크의 상태가 「실행 가능 상태」라면 실행시키는 태스크

### 2. 인터럽트 태스크

인터럽트 요인이 발생하면 상기 싸이클 실행 태스크가 실행중에도 인터럽트를 실행시키는 태스크

인터럽트 태스크는 다음의 4종류가 있습니다. 1) ~ 4) 합계 256개입니다.

1) 전원단 인터럽트 태스크 : 전원단시에 실행 (최대 1개)

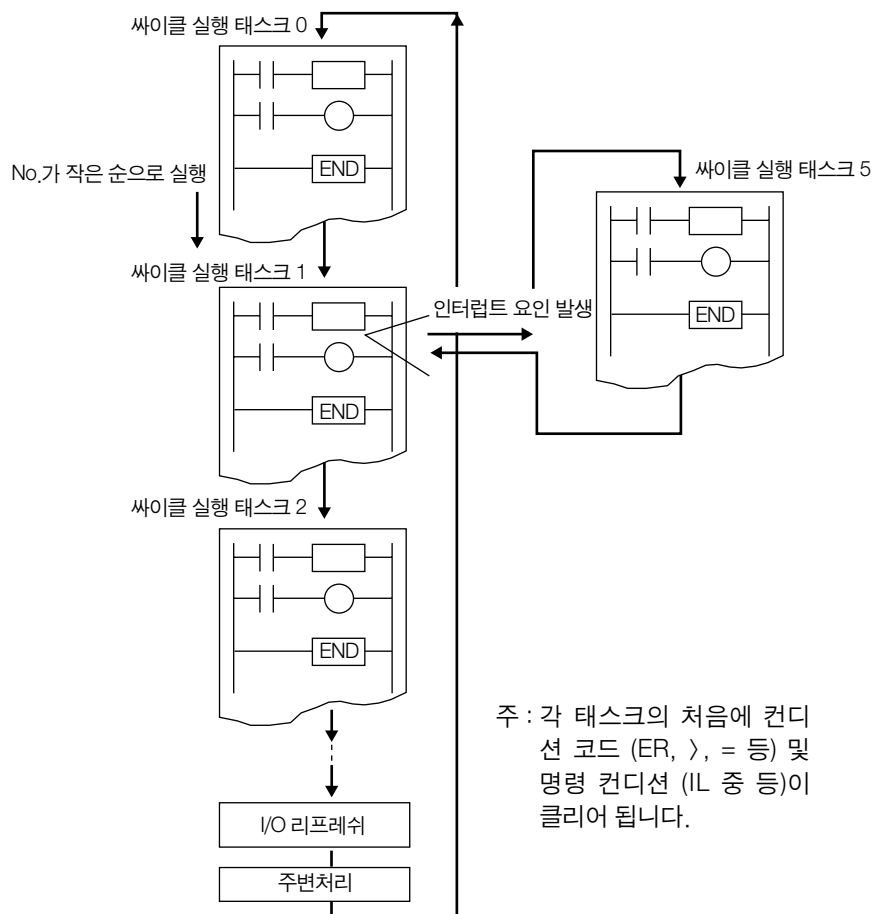
2) 정시 인터럽트 태스크 : 일정시간 간격으로 실행 (최대 2개)

3) I/O 인터럽트 태스크 : 인터럽트 입력 유닛의 접점 기동으로 실행 (최대 32개)

4) 외부 인터럽트 태스크 : 고기능 I/O 유닛, CS1 CPU 고기능 유닛, INNER BOARD로부터 요구가 있을 때 실행 (최대 256개)

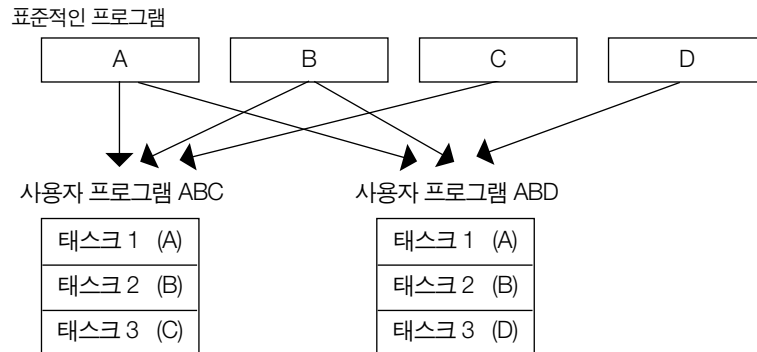
태스크는 상기 합계 최대 288개 (싸이클 실행 태스크 32개 + 인터럽트 태스크 256개) 작성 가능합니다. 프로그램은 최대 288개 작성 · 관리하는 것이 가능합니다(주변 툴 CX-Programmer 사용시).

주변 툴 CX-Programmer의 각 프로그램의 프로퍼티 설정으로 각 프로그램을 태스크에 1=1로 할당합니다.



## 프로그램의 구조화

표준적인 프로그램을 작성해 두고 그것을 필요에 따라 태스크에 할당, 실행용 프로그램을 작성할 수 있습니다. 그것에 의해 프로그램의 모듈(표준부품)화가 가능합니다. 디버그시에도 태스크 별로 디버그를 행할 수 있습니다.



역시, 프로그램을 모듈화 하는 경우 어드레스의 지정은 I/O 명칭으로 지정해 두면 표준화(유용)하기 쉽게 됩니다.

## 프로그램의 실행 / 대기

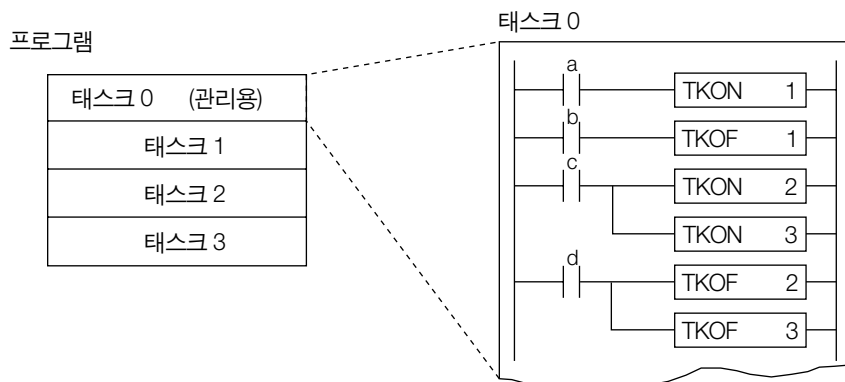
태스크는 태스크 기동 (TKON) 명령/태스크 대기 (TKOF) 명령에 따라 각 태스크가 다른 태스크를 임의로 실행/대기로 하는 것이 가능합니다.

대기상태로 된 태스크는 그 내부의 명령을 실행하지 않습니다만, 입출력 상태는 유지됩니다. 따라서 재실행 상태가 된 때는 이전의 상태에서 재개합니다.

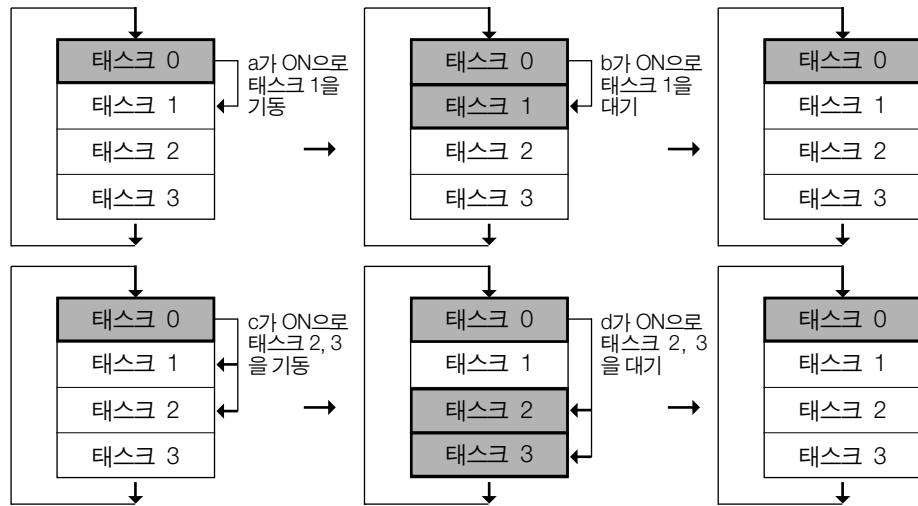
## 예 : 관리용 태스크에 의해 다른 태스크를 제어하는 경우

태스크 0은 관리용 태스크로써 운전개시시에 기동하는 태스크입니다.

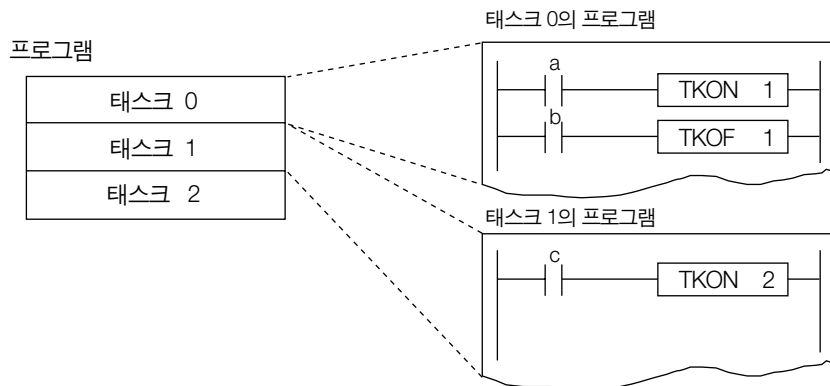
그 밖의 태스크는 주변 톨 (프로콘 제외)의 기능에 의해 설정하는 것으로 운전개시시에 기동하는 것이 가능합니다. 운전 개시후는 태스크 기동 (TKON) 명령/태스크 대기 (TKOF) 명령에 의해 태스크의 상황을 제어 가능합니다.



예 :  
운전개시시, 운전 개시 기동시 설정으로 태스크 0을 실행  
a가 ON시, 태스크 1을 실행  
b가 ON시, 태스크 1을 대기  
c가 ON시, 태스크 2 및 3을 실행  
d가 ON시, 태스크 2 및 3을 대기



예 : 각 태스크가 각 태스크를 제어하는 경우



예 :

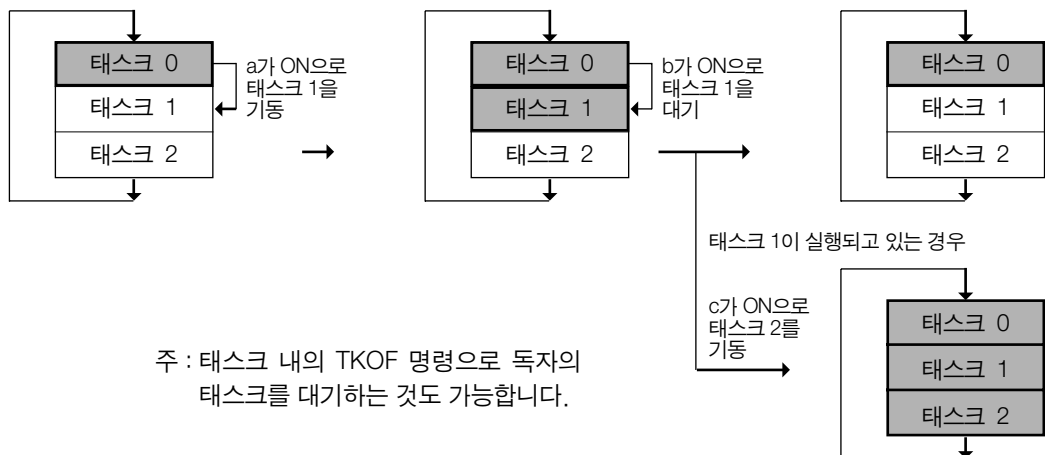
운전개시시 무조건 태스크 0을 실행

a가 ON시, 태스크 1을 실행

b가 ON시, 태스크 1을 대기

태스크 1이 실행되면

c가 ON시, 태스크 2를 실행



주 : 태스크 내의 TKOF 명령으로 독자의 태스크를 대기하는 것도 가능합니다.

## 태스크의 실행시간

태스크가 대기 상태의 경우, 그 중의 명령은 내부적으로도 실행되지 않고 싸이클 타임의 명령 실행 시간으로써 계측되지 않습니다.

주 : 그 점에서 점프 (JMP-JME) 명령과 동일합니다.

따라서 시스템을 전체 관리하는 태스크와 각각의 제어를 하는 태스크로 나뉘서 필요에 따라 어떤 제어 태스크만을 실행하는 것에 의해 시스템 전체의 응답 성능을 향상시키는 것이 가능합니다.

