Real-Time Operating System (Day 4 Lab)

Jong-Chan Kim

Graduate School of Automotive Engineering

KMU 국민대학교 KOOKMIN UNIVERSITY

Cooperative Scheduling

- Non-preemptive 에서 Scheduling Point 설정

```
TASK(Task1)
{
    …
    Schedule();
    …
    Schedule();
    …
}
```

Scheduling Points

- Preemptive 에서 Scheduling Disable

```
TASK(Task1)
{
    …
    GetResource(RES_SCHEDULER);
    …
    ReleaseResource(RES_SCHEDULER);
    …
}
```

No scheduling allowed

Alarm 기반 Activation의 문제점

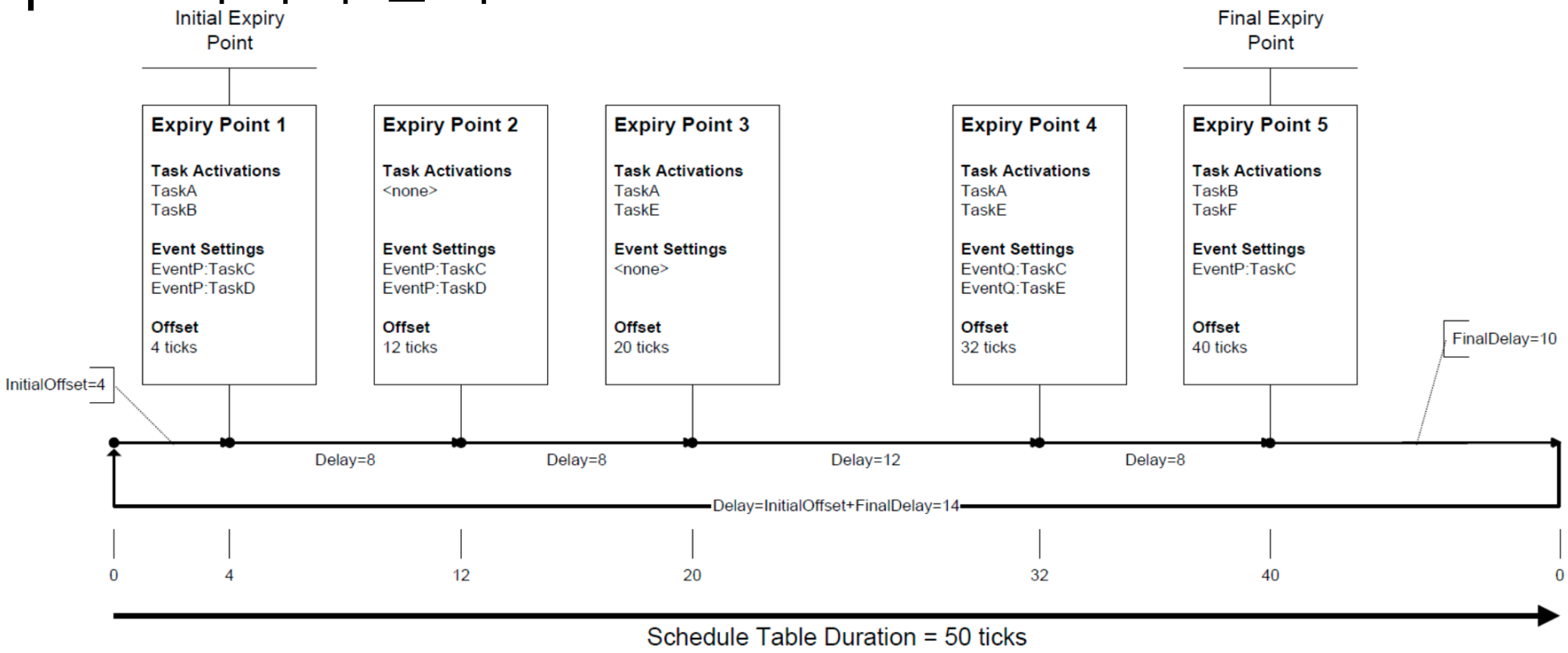
- OSEK에서 Periodic Task 작성 방법
  - Counter와 연결된 Alarm에서 Activate하도록 OIL 설정
- 문제점
  - Alarm은 런타임에 취소/변경이 가능
  - 실수 혹은 악의적으로 Alarm 변경시 Periodic Task 오동작 가능성
- Example)
  - CancelAlarm() 호출

13.6.3.5 CancelAlarm

Syntax:	StatusType CancelAlarm (AlarmType <AlarmID>)
Parameter (In):	
AlarmID	Reference to an alarm
Parameter (Out):	none
Description:	The system service cancels the alarm <AlarmID>.
Particularities:	Allowed on task level and in ISR, but not in hook routines.
	Status:
Standard:	• No error, E_OK
	• Alarm <AlarmID> not in use, E_OS_NOFUNC
Extended:	• Alarm <AlarmID> is invalid, E_OS_ID
Conformance:	BCC1, BCC2, ECC1, ECC2

Schedule Table Concepts

- Duration: Schedule table의 사이클 타임
- Expiry points: Duration 안에서의 상대 시간 (Activate, SetEvent 가능)
- Initial Offset: 첫 expiry point
- Delay: Expiry point 사의의 간격



22. Schedule Table

```
SCHEDULETABLE SchedTab1 {
    COUNTER = counter1;
    DURATION = 10;
    REPEATING = TRUE;
    AUTOSTART = TRUE {
        START_VALUE = 5;
    };
    EXPIRE_POINT = ACTION {
        EXPIRE_VALUE = 0;
        ACTION = ACTIVATETASK { TASK = TaskH; };
        ACTION = ACTIVATETASK { TASK = TaskL; };
    };
    EXPIRE_POINT = ACTION {
        EXPIRE_VALUE = 5;
        ACTION = ACTIVATETASK { TASK = TaskH; };
        ACTION = SETEVENT { TASK = TaskL; EVENT = Event1; };
    };
};
```

Cycle Time

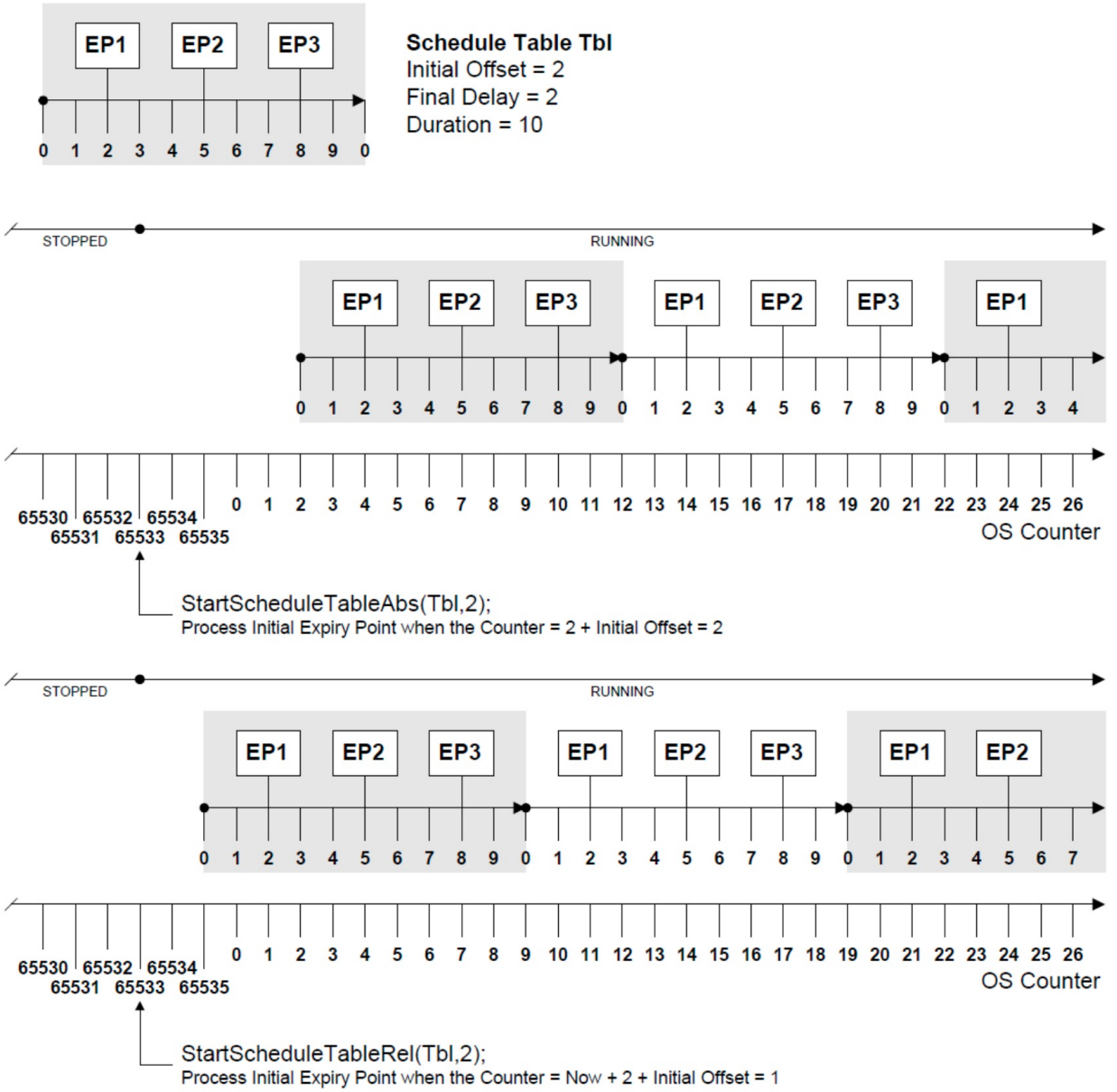
Start Counter Value

Task Activation

Event Setting

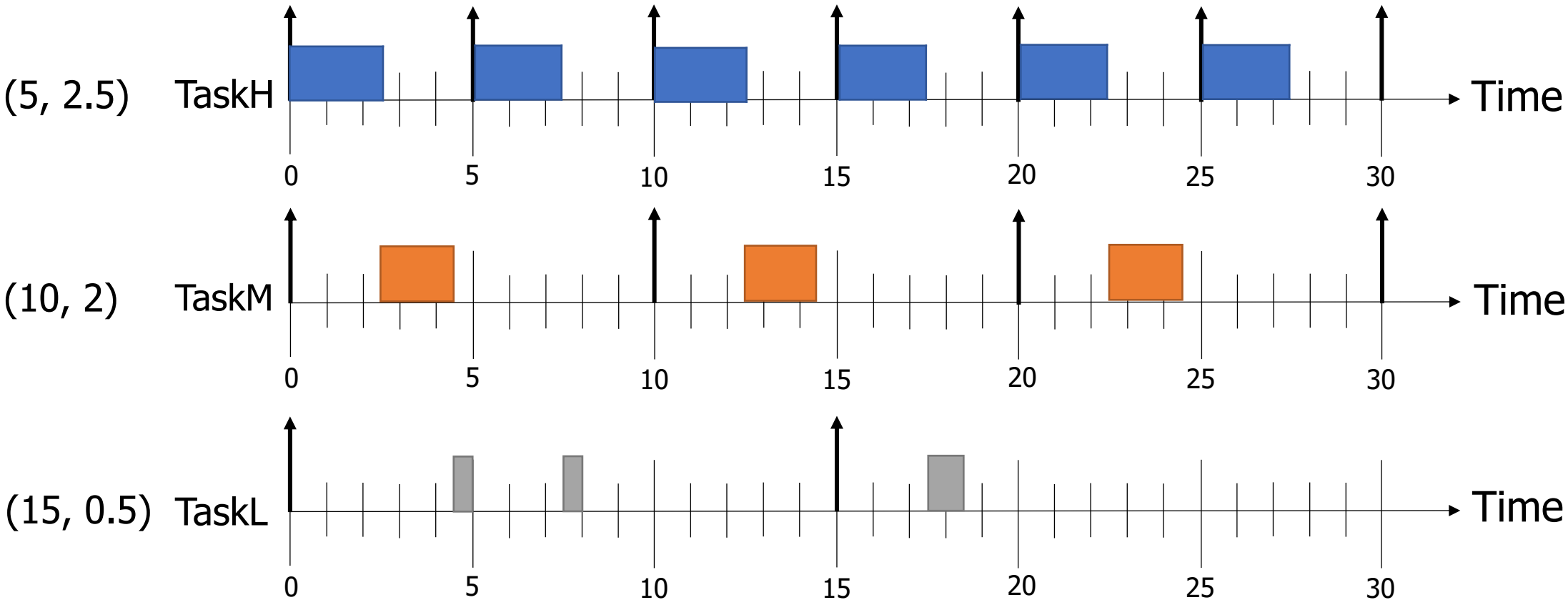
Schedule Table Handling Functions

- StartScheduleTableRel()
- StartScheduleTableAbs()
- StopScheduleTable()



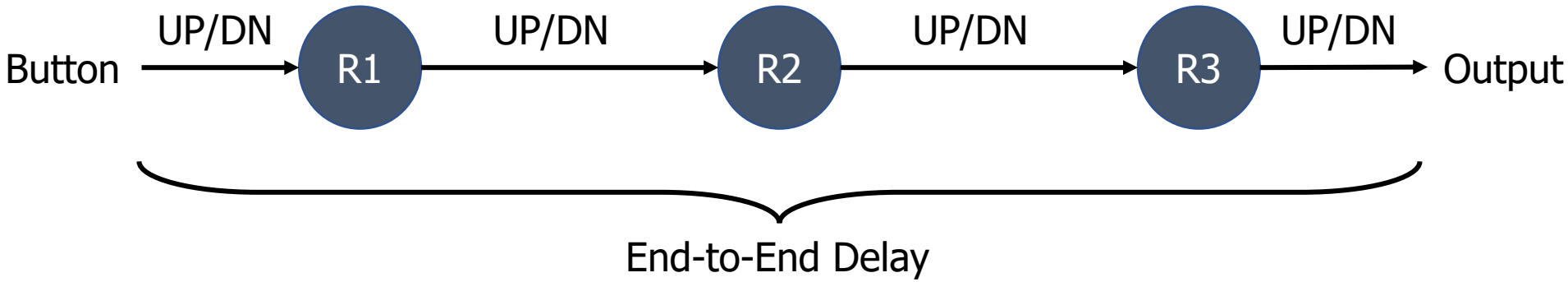
22. Schedule Table

- 아래와 같은 Schedule Table 생성



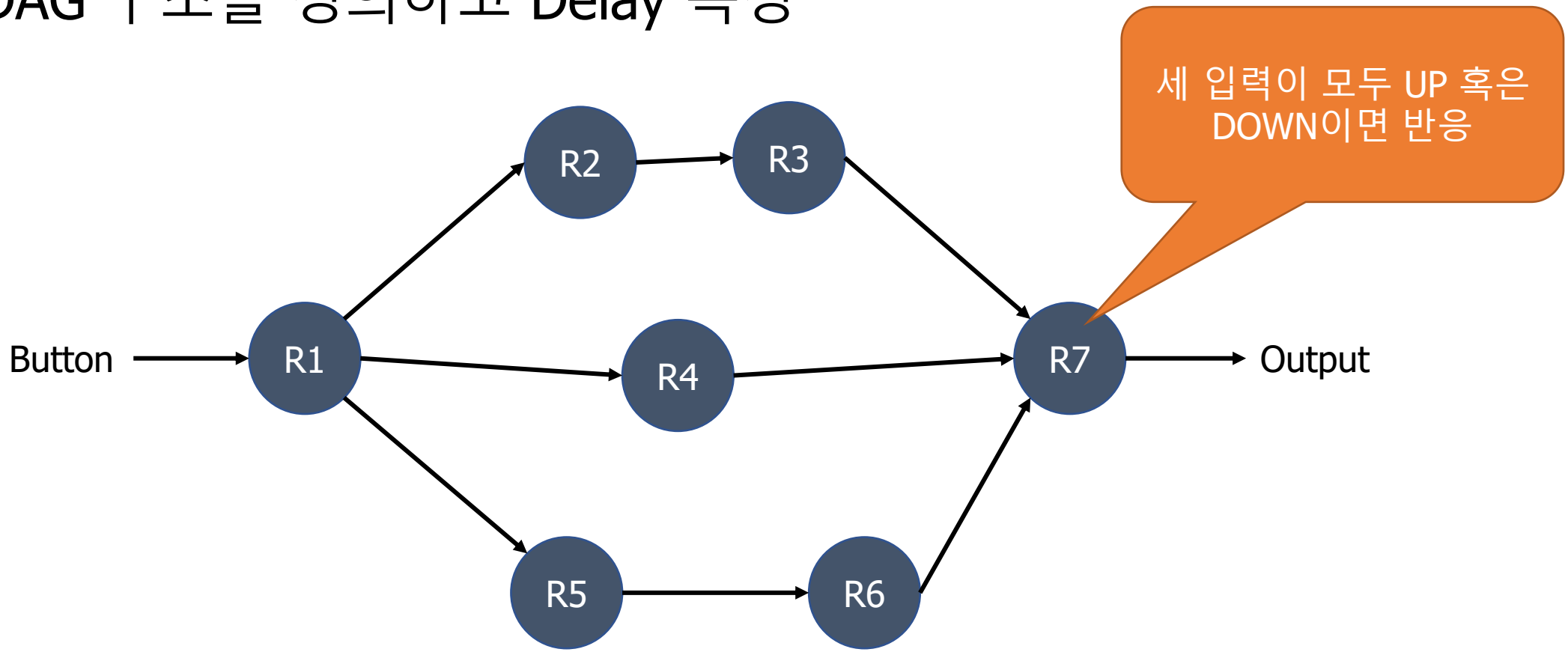
23. End-to-End Delay

- AUTOSAR 기반 DAG (Directed Acyclic Graph) SW 구조
- Runnable to Task 매핑 & 시퀀싱
- Sensor에서 Actuator까지 End-to-End Delay 관찰



23. End-to-End Delay

- 아래 DAG 구조를 정의하고 Delay 측정



Questions