AVR 프로그래밍

5강

정보컴퓨터공학과 권혁동

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

조건문

- 사용할 명령어 모음

명령어	동작
CPSE	값 비교 후 같을 경우 생략
CP	값 비교
CPI	값 즉시 비교
BREQ	같을 경우 분기 형성
BRNE	같지 않을 경우 분기 형성
BRGE	크거나 같을 경우 분기 형성
BRLT	작을 경우 분기 형성
INC	값 증가
RJMP	명령어 점프
CALL	서브루틴 호출

조건문

- C언어에서 if문을 통하여 조건문을 생성
- 어셈블리에서는 각종 **비교 구문을 통해 조건을 형성** 가능
- 다양한 조건문을 사용할 수 있음
- 조건문 형성은 **비교 구문과 분기 구문의 조합**으로 이루어짐

C언어 구문	어셈블리 구문
a != b	CP A, B BRNE xxx
a == b	CP A, B BREQ xxx

조건문

- CPI R18, 6
  - R18의 값을 6과 **즉시 비교**
- BRLT AAA
  - 비교 결과가 작다면, AAA 분기로 이동
  - **그렇지 않다면, 이 명령어는 무시됨**
- RJMP BBB
  - BBB 분기로 이동

```
branch:

MOVW R26, R24

LD R18, X

CPI R18, 6
BRLT AAA
RJMP BBB

AAA:
ADD R18, R18

BBB:

ST X, R18

RET
```

조건문

- 조건문을 형성할 때는 두 가지를 주의
  - **절대 만족하지 못하는** 조건
  - **항상 만족**하는 조건
- 조건문을 응용하는 것으로 반복문 형성 가능

반복문

- 조건문을 응용한 것
- 분기가 반복되는 위치로 이동하면 반복문 완성

코드 설명

- R18의 값을 1 증가
- R18과 10을 비교
- 값이 작다면, LOOP_IN
- 값이 크거나 같다면, LOOP_OUT

```
.global for_loop
.type for_loop, @function

for_loop:

MOVW R26, R24

LD R18, X

LOOP_IN:
INC R18

CPI R18, 10
BRLT LOOP_IN
BRGE LOOP_OUT

LOOP_OUT:

ST X, R18

RET
```

반복문

- BR... 명령어는 명령어와 분기 사이가 너무 멀지 말아야 함
  - 분기 사이에 **63개의 명령어**만 작성 가능
- RJMP를 사용하는 반복문
- RJMP는 분기간 명령어 수에 영향 받지 않음
- 단, CP명령어의 조건을 따지지 않음

```
INC R18
INC R18     //62
INC R18

CPI R18, 10
BRLT LOOP_IN
BRGE LOOP_OUT

LOOP_OUT:

ST X, R18

RET
```

```
INC R18
INC R18     //62
INC R18

CPI R18, 10
RJMP LOOP_IN
BRGE LOOP_OUT

LOOP_OUT:

ST X, R18

RET
```

recipe for target 'ClassApplication.elf' failed

ld returned 1 exit status

반복문

- CALL을 사용한 반복문
- 서브루틴을 형성하는 반복문
- 특정 구문의 실행을 방지하는 효과

```
for_loop:

MOVW R26, R24

LD R18, X

CPI R18, 10
CALL LOOP_IN
BRGE LOOP_OUT

LOOP_OUT:

ST X, R18

RET

LOOP_IN:
INC R18
INC R18
INC R18
INC R18
INC R18
INC R18
INC R18
INC R18
INC R18

RET
```

Q & A