

05장

타겟 머신의 구조

5.7 연습문제

5.16 32비트 2의 보수 산술에서 가장 큰 양수는 무엇인가? 가장 작은(절대 값이 가장 큰) 음수는 무엇인가? 이 둘은 왜 서로 덧셈의 역원이 아닌가?

5.17 (a) 십진수 1234를 16진수로 변환하라.

(b) 부호가 없는 16진수 0x2ae를 십진수로 변환하라.

(c) 16진수 비트 패턴 0xffd9를 16비트 2의 보수로 해석하라. 십진수 값은 무엇인가?

(d) n 이 k 비트 2의 보수 비트 패턴으로 나타낸 음의 정수라고 가정하자. 이 비트 패턴을 부호가 없는 숫자로 다시 해석한다면 n 과 k 의 함수로서의 숫자 값은 무엇인가?

5.18 펜티엄과 같은 리틀 엔디안 기계에서 다음과 같은 C 코드의 출력은 무엇인가? 썬과 같은 빅 엔디안 기계에서의 출력은 무엇인가?

```
unsigned short n = 0x1234;
unsigned char *p = (unsigned char *) &n;
printf ("%d\n", *p);
```

5.19 (a) 하드웨어적으로 8비트 정수를 지원하는 기계가 있다고 가정하자. 부호가 없는 수로 해석할 때 11011001_2 의 십진수 값은 무엇인가? 부호가 있는 수로 해석할 때 2의 보수로는 얼마인가? 2의 보수의 덧셈에 대한 역원은 무엇인가?

(b) 11011001_2 와 10010001_2 의 8비트 이진 합은 무엇인가? 가수(加數)를 부호가 없는 수로 해석하는 경우 합이 오버플로우를 초래하는가? 부호가 있는 2의 보수로 해석하는 경우는 어떻게 되는가?

5.20 2의 보수 덧셈 연산에서 최상위 위치로의 올림과 최상의 위치에서의 올림 값이 다른 경우, 그리고 그 경우에만 오버플로우(결과를 사용할 수 있는 수의 비트에 명시할 수 없는 상황)가 발생한다는 것을 증명하라.

5.21 2의 보수에서 각 비트를 뒤집고 1을 더한 후 가장 왼쪽으로부터의 올림을 모두 무시하는 방법으로 올바르게 해당 2의 보수의 부호를 바꿀 수 있음을 증명하라.

5.22 6.022×10^{23} 에 가장 가까운 단순정밀도 IEEE 부동소수점 실수는 무엇인가?

5.23 가끔 배정도 부동소수점 실수를 정수 카운터로 사용하는 C 프로그램을 보게 된다. 프로그래머는 왜 이렇게 구현했을까?