

Report of Computer Architecture (Simple_Cache_Simulator)

임유택 (201514768)
전북대학교 컴퓨터공학부
j75575863@naver.com

1. 실습 프로그램의 구성 및 동작 원리

trace파일에 있는 값 중 두 번 째 메모리 주소를 이진법으로 나타낸 후 tagBit, indexBit, offsetBit로 나눈다.

```
while (1)
{
    address[position] = totalbit % 2;
    totalbit = totalbit / 2;
    position++;
    if (totalbit == 0) break;
}

for (int i = TOTAL_BIT - 1; i >= 0; i--)
{
    if (i >= TOTAL_BIT - tagBit) tagBitarr[temp++] = address[i];
    if (i == TOTAL_BIT - tagBit) temp = 0;
    if (i < TOTAL_BIT - tagBit && i >= TOTAL_BIT - tagBit - indexBit) indexBitarr[temp++] = address[i];
}
```

그 후 배열에 나뉘져 있는 이진수를 변수 하나에 십진법으로 나타낸다.

```
for (int j = 0; j <= tagBit - 1; j++) inttag = inttag + tagBitarr[tagBit - 1 - j] * pow(2, j);
for (int j = 0; j <= indexBit - 1; j++) intindex = intindex + indexBitarr[indexBit - 1 - j] * pow(2, j);
```

start 변수는 associativity로 나뉜 여러 구역 중 첫 구역에서의 index로 초기화 해 놓은 변수이다.

그렇기 때문에 start에서 indexmax, 즉 index가 가질 수 있는 최대 값인 index gap을 더해주며 tag와 일치하는지 확인한다. 일치한다면 hit를 올리고 Least Recently Used를 구현하기 위해 만들어 놓은 priority라는 배열에 offset만큼 prioritynum를 넣어준다. 그리고 finish로 이동해 다음 tag비교로 넘어간다.

```
for (int s = start; s < cachebyte / cacheblock * offsetBit; s = s + indexmax) {
    for (int a = 0; a < offsetBit; a++) {
        if (cache[s] == inttag) {
            hit++;
            // cout << "hit!!!!!!!!!!!!!!!" << endl;
            for (int b = 0; b < offsetBit; b++) priority[s + b] = prioritynum;
            prioritynum++;
            goto finish;
        }
    }
}
```

만약 끝까지 tag와 메모리가 일치하지 않는다면 이제 index에서 Least Recently Used를 찾아야 하기 때문에 같은 index끼리 접근했던 기록을 비교한다. 같은 index중 최소의 값을 가지고 있는 index를 min으로 저장한다. 그 후 cache[min]에 새로운 tag를 넣어준다.

```

minnum = priority[start];
for (int s = 0; s < associativity; s++) {
    if (priority[start + indexmax * s] <= minnum) {
        minnum = priority[start + indexmax * s];
        min = start + indexmax * s;
    }
}

cache[min] = inttag;
priority[min] = prioritynum;
prioritynum++;

finish::
}

```

예외처리

처음 입력할 때 인자의 수부터 확인한다. 만약 인자의 수가 정상적이라면 프로그램을 작동시킨다.

```

else {
    cout << "In case of not four parameters." << endl;
    return 1;
}

```

그 외의 예외 처리는 입력 받은 cachebyte, cacheblock, associativity로 확인할 수 있다.

```

if (is.fail()) {
    cout << "file does not exist." << endl;
    return 1;
}

double checkcacheblock1 = log2(cacheblock);
int checkcacheblock2 = log2(cacheblock);

if (checkcacheblock1 != checkcacheblock2) {
    cout << "cacheblock is not a exponentiation of 2." << endl;
    return 1;
}

if (!(associativity == 1 || associativity == 2 || associativity == 4 || associativity == 8)) {
    cout << "value of associativity is other than 1, 2, 4 and 8 been entered." << endl;
    return 1;
}

if (cachebyte % (cacheblock * associativity) != 0) {
    cout << "cachesize is not a multiple of cacheblock * associativity." << endl;
    return 1;
}

```

2. 결과

cachebyte 1024, cacheblock 16, associativity 2 기준

```
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 16 2
tag: 23bits
index: 5bits
offset: 4bits
Result: total acces303193, hit 269668, hit rate 0.889426
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim twolf.trace 1024 16 2
tag: 23bits
index: 5bits
offset: 4bits
Result: total acces482824, hit 454532, hit rate 0.941402
```

cachebyte 1024, cacheblock 16, associativity 4 기준

```
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 16 4
tag: 24bits
index: 4bits
offset: 4bits
Result: total acces303193, hit 274979, hit rate 0.906943
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim twolf.trace 1024 16 4
tag: 24bits
index: 4bits
offset: 4bits
Result: total acces482824, hit 460174, hit rate 0.953088
dladbxor@LAPTOP-CONLNHQV:~$
```

cachebyte 1024, cacheblock 16, associativity 8 기준

```
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 16 8
tag: 25bits
index: 3bits
offset: 4bits
Result: total acces303193, hit 276260, hit rate 0.911168
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim twolf.trace 1024 16 8
tag: 25bits
index: 3bits
offset: 4bits
Result: total acces482824, hit 463046, hit rate 0.959036
dladbxor@LAPTOP-CONLNHQV:~$
```

error 처리

```
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 16
In case of not four parameters
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim a.trace 1024 16 2
file does not exist.
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 15 2
cacheblock is not a exponentiation of 2
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1024 16 3
value of associativity is other than 1, 2, 4 and 8 been entered
dladbxor@LAPTOP-CONLNHQV:~$ ./simple_cache_sim swim.trace 1023 16 2
cachesize is not a multiple of cacheblock * associativity
dladbxor@LAPTOP-CONLNHQV:~$
```