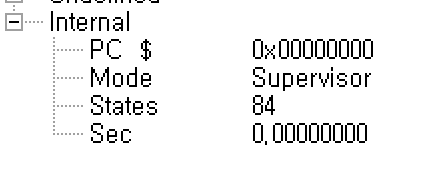
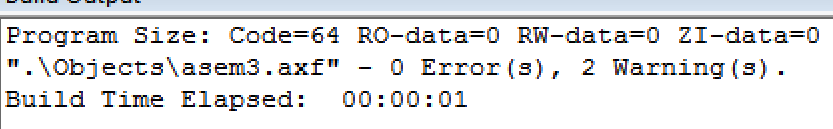
**어셈블리프로그램설계및실습 과제**

**2018202074 김상우**

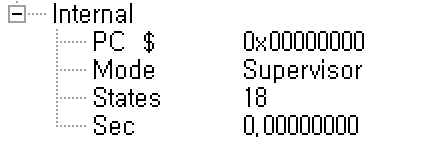
3.1(Loop을 이용한 방법)

아래를 확인하면 알 수 있듯이 Program Size는 64, States는 84가 나온다. Program size는 다른 방법들에 비해 중간이지만 States가 높아 다른 수단들보다 느리다.



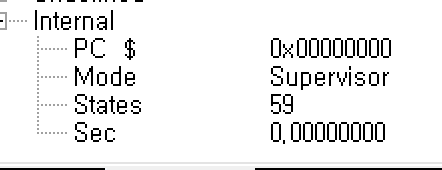
3.2(일반화된 식인 n(n+10)을 이용한 방법)

아래를 확인하면 알 수 있듯이 Program Size는 52, States는 18이 나온다. Program size와 States가 낮아 다른 수단들보다 빠르고 cost가 적어 가장 성능이 좋다고 할 수 있다.



3.3(unlooping을 이용한 방법)

아래를 확인하면 알 수 있듯이 Program Size는 72, States는 59가 나온다. Program size는 다른 수단에 비해 가장 높지만 1번 방법과 비교해 보았을때, States가 낮아 1번 방법보다 빠르게 사용할 수 있다.

위의 결과값들을 표로서 요약하면 다음과 같다.

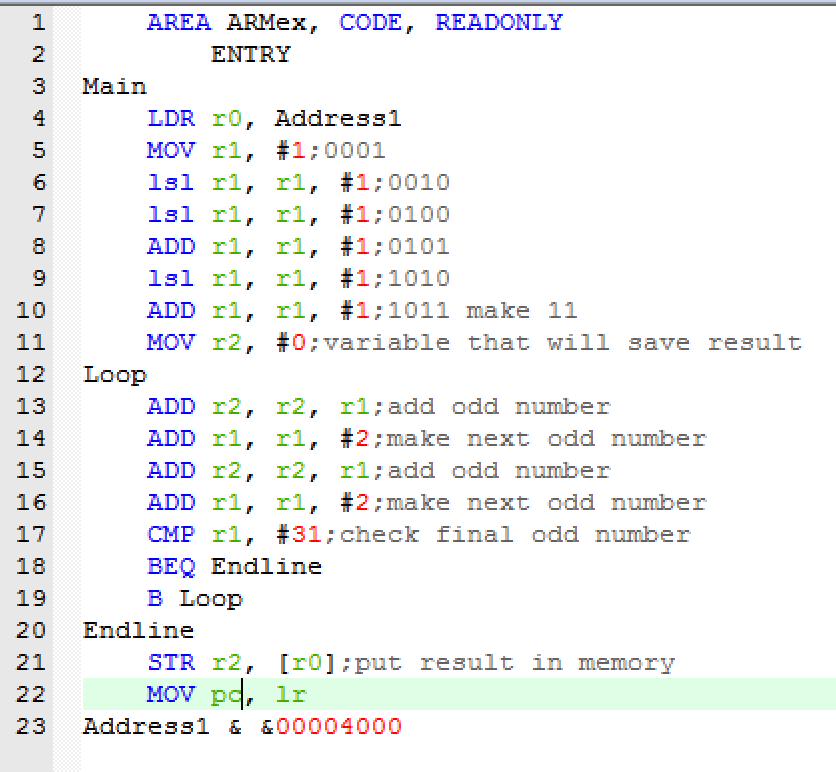
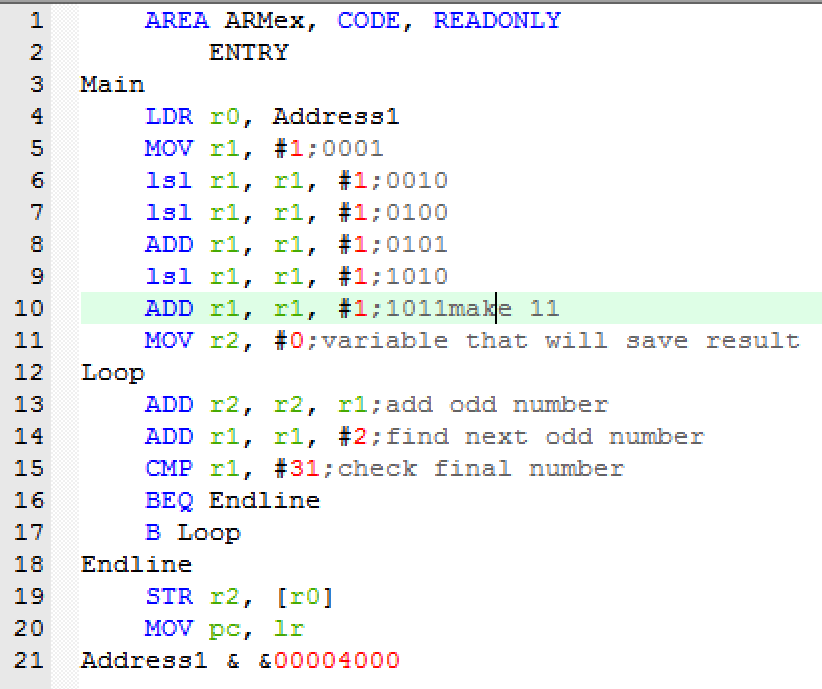
|  |  |  |
| --- | --- | --- |
|  | States | Program size |
| Use Loop | 84 | 64 |
| Use n(n+10) | 18 | 52 |
| Use Unlooping | 59 | 72 |

Use Loop(3-1)와 Use Unlooping(3-3)의 차이가 나는 이유는 Loop의 횟수 때문이다.

Use Loop는 총 10번의 반복을 통해 10번의 덧셈을 10번의 조건확인과 반복을 통해 구한다. 대신 반복 1회당 사용하는 코드가 짧다.(전체 코드는 21줄, 반복되는 코드는 5줄임을 확인할 수 있다.)

반면 Use Unlooping은 10번의 덧셈을 5번의 조건확인과 반복을 통해 구한다. 다만 반복 1회당 2회 덧셈을 취하기 때문에 1회당 사용하는 코드가 길다.(전체 코드는 23줄, 반복되는 코드는 7줄임을 확인할 수 있다.)

그 결과, Use Unlooping의 경우가 더 적은 States를, Use Loop가 더 적은 Program size를 얻게 되었다.



반면 Use n(n+10)을 사용한 경우에 매우 적은 States와 Program size를 가진걸 알 수 있는데, 이는 MUL을 통해 조건 확인과 Loop를 하지 않게 되어, 다음과 같은 결과가 나온 것이다.

