12161633 이영주

(첫번째 Loop)

ADD START 0000 // 주소 00000000 인 ADD라는 이름에서 시작

LDX ZERO // 목적코드: 04(16), ZERO를 X 레지스터로 가져옴

000001 110010 000000100100(072024)

X레지스터: 0 저장, PC는 LDA FIVE를 가리키고

있고, p=1이므로 pc relative한 주소방식을 가지고

있다. 1 word는 3 byte이므로 PC + 36은

PC로부터 12칸 떨어져 있는 ZERO WORD 0을

가리킴.

LDA FIVE // 목적코드 :00(16), FIVE를 A 레지스터로 가져옴

000000 110010 000000100100(032024)

A 레지스터: 5 저장, PC는 STA ALPHA를 가리키고있고, p=1이므로 pc relative하다. PC + 36은 PC로부터 12칸 떨어진 FIVE WORD 5를 가리킴.

STA ALPHA // 목적코드 :0C(16), A레지스터에 있는 값을

ALPHA에 저장

000011 110010 000000100100(0F2024)

레지스터 값들은 안 바뀜, PC는 STA BETA를 가리킴. p=1이므로 pc relative한 주소지정방식이다. PC+36은 현재 PC로부터 12칸 떨어져 있는 ALPHA RESW 1을 가리킴 ALPHA에는 A레지스터에 저장되어 있던 값 5가 복사됨

STA BETA // 목적코드 :0C(16), A레지스터에 있는 값을

BETA에 저장

000011 110010 000000100100(0F2024)

이 명령어 또한 레지스터에 저장되어 있는 값은 바뀌지 않음, PC는 MOVE LDA ALPHA를 가리킴 p=1이므로 pc relative하다. PC+36은 PC로부터 12칸 떨어져 있는 BETA RESW 1을 가리킴 BETA에는 A레지스터에 저장되어 있던 값 5가 복사됨

MOVE LDA ALPHA // 목적코드 :00(16), A 레지스터에 ALPHA를 저장

000000 110010 000000011110(03201E)

A 레지스터 : 5 저장, PC는 ADD FIVE를 가리키고, p=1 이므로 pc relative한 주소형태이다. PC+30은 현재 PC로부터 10칸 떨어져 있는 ALPHA RESW 1을 가리킴

ADD FIVE // 목적코드 :18(16), A레지스터 값에 FIVE를 더함

000110 110010 000000011000(1B2018)

A 레지스터 : A 저장, PC는 STA ALPHA를 가리키고, p=1이므로 pc relative하다. PC+24는 PC로부터 8칸 떨어져 있는 FIVE WORD 5를 가리킨다. 원래 저장되어 있던 5에 5를 더하여 00000A가 된다.

STA ALPHA // 목적코드 :0C(16), A 레지스터 값을 ALPHA에

저장

000011 110010 000000011000(0F2018)

레지스터 값들은 변하지 않음. PC는 LDA BETA를 가리키고, p=1 이므로 pc relative하다. PC+24는 PC로부터 8칸 떨어져 있는 ALPHA RESW 1을 가리킨다. ALPHA에는 00000A값이 저장된다.

LDA BETA // 목적코드 :00(16), A 레지스터에 BETA를 저장 000000 110010 000000011000(032018)

A 레지스터 : 5, PC는 ADD ALPHA를 가리킴.

p=1이므로 pc relative한 주소형태, PC + 24이므로 PC에서부터 8칸 떨어져 있는 BETA RESW 1을 가리킴.

ADD ALPHA // 목적코드 :18(16), A 레지스터 값에 ALPHA를 더함

000110 110010 000000010010(1B2012)

A 레지스터 : F,PC는 STA BETA를 가리킴. p=1이므로 pc relative함. PC+18은 PC로부터 6칸 떨어져 있음을 의미함, ALPHA RESW 1을 가리킴. ALPHA에 저장되어 있는 값을 A레지스터 값에 더함

STA BETA // 목적코드 :0C(16), A 레지스터 값을 BETA에 저장

000011 110010 000000010010(0F2012)

레지스터 값들은 변하지 않는다. PC는 다음 순서인 TIX FIVE를 가리킨다. p=1이므로 pc relative한 주소지정방식을 가지고 있다. PC + 18이므로 PC로부터 6칸 떨어진 BETA RESW 1을 가리킨다. A 레지스터에 저장된 00000F를 BETA에 저장한다.

TIX FIVE // 목적코드 :2C(16), X레지스터에 저장된 값을 1만큼

증가시킨 후, X값과 FIVE를 비교하여 결과를 SW

레지스터에 저장

001011 110010 000000001001(2F2009)

X 레지스터 :1, SW 레지스터 : 000040로 변환, PC는

JLT MOVE를 가리킴. PC + 9이므로 현재 PC에서

3칸 떨어진 FIVE WORD 5와 X 레지스터 값을

비교한다.

JLT MOVE // 목적코드 :38(16), MOVE가 SW 레지스터에

저장되어 있는 값보다 작으면 PC의 위치를 MOVE로 옮긴다.

001110 110010 111111101000(3B2FE8)

레지스터 값은 변화가 없다. MOVE가 SW 레지스터보다 작으므로 PC는 MOVE가 있는 00000C로 이동한다.

RSUB

.

ZERO WORD 0

FIVE WORD 5

ALPHA RESW 1

BETA RESW 1

END ADD

(마지막 Loop)

MOVE LDA ALPHA // 목적코드 :00(16), A 레지스터에 ALPHA를 저장

000000 110010 000000011110(03201E)

A 레지스터 : 19 저장, PC는 ADD FIVE를 가리키고, p=1 이므로 pc relative한 주소형태이다. PC+30은 현재 PC로부터 10칸 떨어져 있는 ALPHA RESW 1을 가리킴

ADD FIVE // 목적코드 :18(16), A레지스터 값에 FIVE를 더함

000110 110010 000000011000(1B2018)

A 레지스터 : 1E 저장, PC는 STA ALPHA를 가리키고, p=1이므로 pc relative하다. PC+24는 PC로부터 8칸 떨어져 있는 FIVE WORD 5를 가리킨다. 원래 저장되어 있던 5에 5를 더하여 00001E가 된다.

STA ALPHA // 목적코드 :0C(16), A 레지스터 값을 ALPHA에

저장

000011 110010 000000011000(0F2018)

레지스터 값들은 변하지 않음. PC는 LDA BETA를 가리키고, p=1 이므로 pc relative하다. PC+24는 PC로부터 8칸 떨어져 있는 ALPHA RESW 1을 가리킨다. ALPHA에는 00001E값이 저장된다.

LDA BETA // 목적코드 :00(16), A 레지스터에 BETA를 저장 000000 110010 000000011000(032018)

A 레지스터 : 4B, PC는 ADD ALPHA를 가리킴.

p=1이므로 pc relative한 주소형태, PC + 24이므로 PC에서부터 8칸 떨어져 있는 BETA RESW 1을 가리킴.

ADD ALPHA // 목적코드 :18(16), A 레지스터 값에 ALPHA를 더함

000110 110010 000000010010(1B2012)

A 레지스터 : 69,PC는 STA BETA를 가리킴. p=1이므로 pc relative함. PC+18은 PC로부터 6칸 떨어져 있음을 의미함, ALPHA RESW 1을 가리킴. ALPHA에 저장되어 있는 값을 A레지스터 값에 더함

STA BETA // 목적코드 :0C(16), A 레지스터 값을 BETA에 저장

000011 110010 000000010010(0F2012)

레지스터 값들은 변하지 않는다. PC는 다음 순서인 TIX FIVE를 가리킨다. p=1이므로 pc relative한 주소지정방식을 가지고 있다. PC + 18이므로 PC로부터 6칸 떨어진 BETA RESW 1을 가리킨다. A 레지스터에 저장된 000069를 BETA에 저장한다.

TIX FIVE // 목적코드 :2C(16), X레지스터에 저장된 값을 1만큼

증가시킨 후, X값과 FIVE를 비교하여 결과를 SW

레지스터에 저장

001011 110010 000000001001(2F2009)

X 레지스터 :5, SW 레지스터 : 000000로 변환, PC는

JLT MOVE를 가리킴. PC + 9이므로 현재 PC에서

3칸 떨어진 FIVE WORD 5와 X 레지스터 값을

비교한다.

JLT MOVE // 목적코드 :38(16), MOVE가 SW 레지스터에

저장되어 있는 값보다 작으면 PC의 위치를 MOVE로 옮긴다.

001110 110010 111111101000(3B2FE8)

레지스터 값은 변화가 없다. MOVE가 SW 레지스터보다 크므로 PC는 그 다음 문장인 00021인 RSUB로 이동한다.

RSUB // 리턴

.

ZERO WORD 0 // 변수 ZERO에 0을 저장

000000

FIVE WORD 5 // 변수 FIVE에 5를 저장

000005

ALPHA RESW 1 // 변수 ALPHA에 1 WORD크기만큼을 미리 할당

BETA RESW 1 // 변수 ALPHA에 1 WORD크기만큼을 미리 할당

END ADD // 프로그램 종료