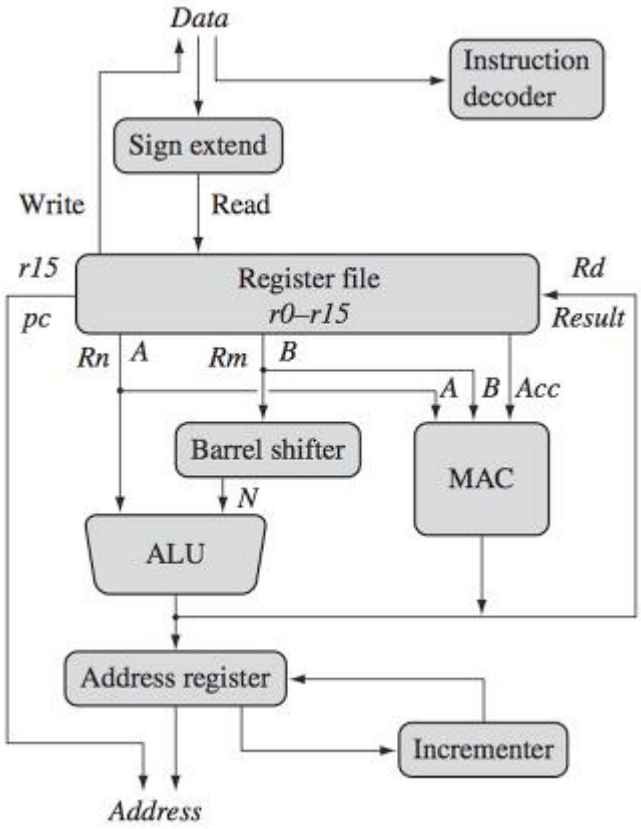
# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 최대성
c3d4s19@naver.com

2018.04.30. – 044일차 수업

---

ARM 어셈블리어 수업




ARM – 폰노이만 구조(데이터와 명령어가 버스 공유)

<–> 하버드 구조(데이터와 명령어용 버스 분리)

Data

-> Data Bus를 통해 core로 입력

Instruction decoder

-> 명령어 실행 전 해독

MAC 곱셈기

-> 옵션으로 들어있는 하드웨어이고 곱셈 + 덧셈을

최대 4개까지 동시에 병렬 수행하여 1clock에 처리 가능함

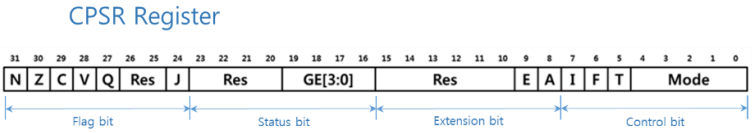
있으면 성능 대폭 증가함

전체적인 내용 아래 링크 참고

http://jiming.tistory.com/183?category=577510

---

CPSR(Current Program Status Register)




-> 현재 프로그램 상태 저장하는 레지스터

아래 링크 참고

http://forum.falinux.com/zbxe/index.php?document_srl=575677&mid=lecture_tip

---

*ARM 어셈블리 함수

add r3, r1, r2

(add)

-> r3 = r1 + r2

subgt r3, r1, r2

(subtract)

-> r3 = r1 - r2

rsble r3, r1, r2

(reverse subtract)

-> r3 = r2 - r1

and r3, r1, r2

-> r3 = r1 & r2

biceq r3, r1, r2

(bit clear)

-> r3 = r1 & ~r2

orr r3, r1, r2

-> r3 = r1 | r2

eors r3, r1, r2

(exclusive or)

-> r3 = r1 ^ r2

mov r1, r2

-> r1 = r2

mvn r1, r2

-> r1 = 0xffffffff ^ r2

cmp r1, r2

-> r1 - r2 하여 비교 후 state flag 업데이트

cmn r1, r2

(cmp negative)

-> r1 + r2 하여 비교

tsteq r1, r2

-> r1 & r2 하여 비교 후 조건 flag 업데이트

teq r1, r2

-> r1 ^ r2 하여 비교 후 조건 flag 업데이트

---

산술 명령어 뒤에 접미사 gt ge lt le eq 의미 (status flag 상태를 보고 실행할지 말지 결정)

gt -> great than

ge -> great equal

lt -> less than

le -> less equal

eq -> equal

ne -> not equal

산술 명령어 뒤에 접미사 s 가 붙을 경우

-> 산술 처리 결과에 따라 flag 업데이트

---

sudo apt-get update

sudo apt-get install qemu-user-static qemu-system

sudo apt-get install gcc-arm-linux-gnueabi

이후에 아무런 C 소스 파일을 작성한다.

arm-linux-gnueabi-gcc -g 소스파일

sudo apt-get install gdb-multiarch

터미널을 2개 띄운다.

A 터미널에서 아래 명령어를 수행한다.

qemu-arm-static -g 1234 -L /usr/arm-linux-gnueabi ./a.out

B 터미널에서 아래 명령어를 수행한다.

gdb-multiarch

file a.out

target remote localhost:1234

b main

c

이후부터 디버깅을 진행하면 된다.

다른 모든 프로그램도 이러한 방법으로 디버깅을 수행할 수 있다.

이제 ARM Architecture에 대해 Deep Down 할 수 있는 기점이 마련되었다.

출처 링크

http://cafe.naver.com/hestit/3204

---