

분기 나누기

분기명령

ble – 분기 명령

ble.s

ble.un

ble.un.s

blt – 분기 명령

blt.s

blt.un.s

bne.un (bne) – 조건이 같지 않을 경우 분기

bne.un.s

br 분기

br.s

Jmp

Ja - 결과가 1보다 클 경우 점프

Jb - 결과가 1보다 작을 경우 점프

Jbe - 결과가 1보다 작거나 같을 경우 점프

Jecxz - ecx 레지스터가 0일때 점프

Jg - 결과가 클 경우 점프(signed)

Jge - 결과가 크거나 같을 경우 점프 (signed)

Jl - 결과가 작을 경우 점프(signed)

Jle - 결과가 작거나 같을 경우 점프 (signed)

jmp

Jnb - 작지 않을 때 점프

Jno - 오버플로 플래그가 1일 때 / 0일 때 점프

Jnp - 패리티 플래그가 0일 때 / 홀수일 때 점프

Jns - 사인(부호) 플래그가 1일 때(음수) / 0일 때(양수) 점프

Jnz - 결과가 0이 아닐 때 점프

Jo - 오버플로 플래그가 1일 때 / 0일 때 점프

Jp - 패리티 플래그가 1일 때 / 짝수일 때 점프

Js - 사인(부호) 플래그가 1일 때(음수) / 0일 때(양수) 점프

Jz - 결과가 0일 때 점프

플래그 설정

cmp - 상태 플래그를 설정

Cmpsb – 비교 명령어

cmpsd - 부동 소수점 값을 비교하고 비교 결과를 대상 피연산자와 비교

cmpxchg - AL, AX, EAX 또는 RAX 레지스터의 값을 비교

cmpxchg8b

comisd - CF 플래그를 설정

comiss - CF 플래그를 설정

switch

연산

연산- aad, and

Aaa – 덧셈 2자리

aad – 나눗셈 작업

aam - BCD쌍을 곱셈

aas – 뺄셈

adc – 캐리 플래그 덧셈

Add – 덧셈

add.ovf

Addpd – 덧셈한 부동소수점

addps – 부동소수점 결과

Addsd – 부동소수점 결과 저장

Adds - 부동소수점 결과 저장

and – 논리연산자 and

andnps – 논리 연산자 and not

andpd – 논리 연산 and

andps – 비트 논리 and

연산 – div

div – 나눗셈

divsd – 나눗셈 결과 저장

divss – 나눗셈 결과 저장

부동소수점 – 연산

fdiv – 부동소수점 나눗셈

fdivp

fdivr

fdivrp

Fsub – 부동소수점 나눗셈

fsubp

fsubr

fsubrp

Fimul – 부동소수점 곱셈

연산 – 빼기

sub – 빼기

sub.ovf

subpd - 팩형 배정 밀도 부동 소수점 값 빼기

subps - 팩형 단 정밀도 부동 소수점 값 빼기

subsd - 스칼라 배정 밀도 부동 소수점 값 빼기

subss - 스칼라 단 정밀도 부동 소수점 값 빼기

idiv – 나눗셈 후 저장

imul - 두 피연산자의 부호 있는 곱을 수행

부동소수점 – 비교

Fcom - 부동 소수점 값 비교

fcomi

fcomip

fcomp

fcompp

fucom 부동소수점 부호없는 비교

fucomi

fucomip

fucomp

fucompp

논리 연산자

pand – 논리 and

pandn – 논리 및 NOT

pxor - 논리 배타적 OR

vmulsd

xor - 논리 배타적 OR

vmulss

xorpd - 팩형 배정도 부동 소수점 값의 비트 단위
논리 XOR

xorps - 팩형 단 정밀도 부동 소수점 값의 비트 단
위 논리 XOR

xor - 논리 배타적 OR

xorpd - 팩형 배정도 부동 소수점 값의 비트 단위
논리 XOR

xorps - 팩형 단 정밀도 부동 소수점 값의 비트 단
위 논리 XOR

연산 - 곱하기

mul - 부호없는 곱하기

mul.ovf.un

mulpd - 다중 팩 배정도 부동 소수점 값 곱하기

mulps - 다중 팩된 단 정밀도 부동 소수점 값 곱하
기

mulsd - 스칼라 배정 밀도 부동 소수점 값 곱하기

mulss - 스칼라 단 정밀도 부동 소수점 값 곱하기

neg - 피연산자 값 (대상 피연산자)을 2의 보수로
바꿉니다.

sbb - 차용자와 정수 빼기

fmul - 곱셈

not - not 연산

or - or 연산

Orps - 팩형 단일 정밀 부동 소수점 값의 비트 단위
논리 OR

vmulpd

vmulps

메모리 관련

Mov

mov – move

movapd - 2, 4 또는 8 배정 밀도 부동 소수점 값을
mov

movaps - 4, 8 또는 16의 단 정밀도 부동 소수점 값
mov

movd

movdqa

movdqu

movhlps - 패킹 된 단 정밀도 부동 소수점 값을 높
음에서 낮음으로 이동

movhpd - 높은 배정도 부동 소수점 값 이동

movhps - 밀도 단 정밀도 부동 소수점 값 이동

movlhps - 패킹 된 단 정밀도 부동 소수점 값을 낮
음에서 높음으로 이동

movlpd - 낮은 팩 배정도 부동 소수점 값 이동

movlps - 낮은 수준의 단 정밀도 부동 소수점 값
이동

movq - 쿼드 워드 이동

movsb – 바이트단위로 반복해서 옮김

movsd - 스칼라 배정 밀도 부동 소수점 값 이동 또
는 병합

movss - 스칼라 단 정밀도 부동 소수점 값 이동 또
는 병합

movsw – string to string

movsx – 더블워드 목적지에 전송

movupd - 정렬되지 않은 팩 배정도 부동 소수점
값 이동

movups - 정렬되지 않은 패킹 된 단 정밀도 부동

소수점 값 이동

movzx - 제로 확장으로 이동

vmovaps - 정렬된 팩형 단 정밀도 부동 소수점 값
이동

vmovd – 더블 워드 이동

vmovdqa

vmovdqu

vmovmskps

vmovq

vmovsd

vmovss

vmovupd

vmovups

push

push - 스택에 워드, 더블 워드 또는 쿼드 워드를
푸시

pusha

pushf

스택

ldarg.0 - 지정된 인덱스 값이 참조하는 인수를 스택
으로 로드

ldarg.1

ldarg.2

ldarg.3

ldarg.s

ldarga.s

ldc.i4 - 메모리에서 보조 프로세서로 데이터를 전송

ldc.i4.0	ldind.i1
ldc.i4.1	ldind.i2
ldc.i4.2	ldind.i4
ldc.i4.3	ldind.i8
ldc.i4.4	ldind.r4
ldc.i4.5	ldind.r8
ldc.i4.6	ldind.ref
ldc.i4.7	ldind.u1
ldc.i4.8	ldind.u2
ldc.i4.m1	ldind.u4
ldc.i4.s	ldlen - 1차원 배열의 요소 수를 계산 스택으로 푸시합니다
ldc.i8	
ldc.r4	ldloc - 지역 변수를 계산 스택으로 로드합니다.
ldc.r8	ldloc.0
ldelem - 배열 요소를 계산 스택 맨 위에 로드	ldloc.1
ldelem.i4	ldloc.2
ldelem.i8	ldloc.3
ldelem.r4	ldloc.s
ldelem.ref	ldloca - 특정 인덱스에 있는 지역 변수의 주소를 계산 스택으로 로드
ldelem.u1	ldloca.s
ldelem.u2	ldnull - null 참조(O 형식)를 계산 스택으로 푸시합니다.
ldelem.u4	
ldelema - 배열 요소의 주소를 계산 스택 맨 위에 로드	ldobj - 주소가 가리키는 값 형식 개체를 계산 스택 맨 위로 복사
ldfld - 현재 계산 스택에 참조가 있는 개체에서 필드의 값을 찾습니다.	lds
ldflda	ldsfld - 정적 필드의 값을 계산 스택으로 푸시합니다.
ldftn - (native int 형식)를 계산 스택으로 푸시	ldsfla - 정적 필드의 주소를 계산 스택으로 푸시

ldstr - 새 개체 참조를 메타데이터에 저장된 문자열 리터럴로 푸시합니다.

Ldtoken - 메타데이터 토큰을 런타임 표현으로 변환하여 계산 스택으로 푸시합니다.

Ldvirtftn - 가상 메서드를 평가 스택에 구현하는 네이티브 코드로 푸시

Ceq - 두 값이 같으면 스택으로 1 다르면 0 푸시

Cgt - 첫째 값이 둘째 값보다 크면 1 아니면 0 스택으로 푸시

localloc - 계산 스택으로 푸시

pop - 스택에서 value 추출

popa - 스택에 있는 값을 레지스터에 복원

popf - 스택 맨 위에서 단어를 길게 표시하고 값을 플래그 레지스터에 저장

fild - FPU 레지스터 스택으로 푸시

fldz - stack에 0을 push

rem - 두 값을 나누어 나머지를 계산 스택으로 푸시

return

비교

maxps – 최대 값 반환

maxsd – 최대값을 하위 쿼드 워드로 반환

maxss - 하위 더블 워드에 최대 값을 반환

minps

minsd - 하위 쿼드 워드에 최소값을 반환

Minss - 하위 더블 워드에 최소값을 반환

ret – return 복귀

retn

retf

etc

패킹

packssdw

packuswb -

paddb

paddd

paddsw

paddusw

paddw

pmaddusw - 팩형 부호 및 부호없는 바이트 곱하기 및 추가

pmaddwd - 팩형 정수 곱하기 및 추가

pmaxsw

pminsw

pmullw - 섀미 부호있는 정수 및 낮은 결과 저장

psadbw - 절대 차이의 계산 합계

pshufb - 팩 셔플 바이트

pshufd - 셔플 팩 더블 워드

pshufw - 셔플 팩 낮은 단어

pshufw - 단어 섞기

pslld

psllq

psllw

psrad

psraw

psrld

psrldq - 더블 워드 워드 오른쪽 논리 시프트

psrlq

psrlw

psubb

psubd

psubusb

psubw

punpckhbw

punpckhdq

punpckhwd

punpcklbw

punpckldq

punpcklqdq

punpcklwd

unpcklpd - 낮은 팩킹 배정도 부동 소수점 값의 압축을 풀고 인터리브

unpcklps - 압축이 낮은 단 정밀도 부동 소수점 값의 압축을 풀고 인터리브

vaddpd

vaddps

vaddsd

vaddss

vandpd

vandps

daa - 압축 된 두 BCD 값의 합계를 조정

das - 압축 된 BCD 값 두 개를 뺀 결과를 조정

set

setalc

setb

setbe	stind.i4
setl	stind.i8
setle	stind.r4
setnb	stind.r8
setnbe	stind.ref
setnl	stloc - 계산 스택 맨 위에서 현재 값을 팝하고 지정된 인덱스에 있는 지역 변수 목록에 저장
setnle	
setns	stloc.0
setnz	stloc.1
seto	stloc.2
sets	stloc.3
setz	stloc.s
stc - 캐리 플래그 설정	stobj - 지정된 형식의 값을 계산 스택에서 주어진 메모리 주소로 복사
std - 방향 플래그 설정	stosb - 지정 주소에 저장
stelem - 지정된 인덱스에 있는 배열 요소를 명령에 지정된 형식을 갖는 계산 스택의 값으로 바꿉니다.	stosd
stelem.i1	stosw
stelem.i2	stsfld
stelem.i4	call
stelem.i8	call - 스택 및 분기에 대한 정보를 연결하는 프로시저를 대상 피연산자를 사용하여 지정된 호출된 프로시저에 저장합니다.
stelem.r4	
stelem.ref	Calli - 호출 규칙에서 설명하는 인수를 사용하여 계산 스택에 표시된 메서드를 진입점에 대한 포인터로 호출합니다.
stfld - 개체 참조나 포인터의 필드에 저장된 값을 새 값으로 바꿉니다.	Callvirt - 개체에서 런타임에 바인딩된 메서드를 호출하고 반환 값을 계산 스택으로 푸시합니다.
sti - 인터럽트 플래그 설정	조건부 대입
stind.i - 주어진 주소에 native int 형식의 값을 저장	Cmov - cmov : 조건부 대입
stind.i1	cmovb
stind.i2	

cmovbe

cmovg

cmovge

cmovl

cmovle

cmovnb

cmovns

cmovnz

cmovs

cmovz

부동소수점으로 변환

cvtdq2pd - 부동 소수점 값으로 변환

cvtdq2ps - 부동 소수점 값으로 변환

cvtpd2ps - 부동 소수점 값으로 변환

cvtps2dq - 더블 워드 정수로 변환

cvtps2pd - 부동 소수점 값으로 변환

cvtsd2ss - 부동 소수점 값으로 변환

cvtsi2sd - 부동 소수점 값으로 변환

cvtsi2ss - 부동 소수점 값으로 변환

cvtss2sd - 부동 소수점 값으로 변환

cvtsd2si - 부호있는 쿼드 워드 정수로 변환

cvtss2si - 부호있는 쿼드 워드 정수로 변환합니다.

arpl - 두 세그먼트 선택기의 RPL 필드를 비교합니다.

bound - 지정된 배열의 범위 내에 있는지 여부를 결정합니다

box

brfalse

brfalse.s

brtrue - value가 true이거나 null이 아니거나 0이 아닌 경우 대상 명령으로 제어를 전달합니다.

brtrue.s

bsf - 최하위 세트 비트 (1 비트)를 검색

Bsr - 소스 피연산자 (두 번째 피연산자)를 검색

bswap - 바이트 순서를 반대로 바꿈

bt - 비트 값을 CF 플래그에 저장

btr - 비트 값을 CF 플래그에 저장하고 선택한 비트를 지움

bts - 비트 값을 CF 플래그에 저장하고 선택된 값을 설정

Castclass - 참조로 전송된 개체를 지정된 클래스로 캐스팅

Cdq - 나눗셈을 위해 사이즈를 확장

cgt.un

clc - EFLAGS 레지스터에서 CF 플래그를 지움

cld - EFLAGS 레지스터에서 DF 플래그를 지움

cli - IF 플래그를 지움

clt

cmc - EFLAGS 레지스터에서 CF 플래그를 보완합니다.

constrained.

conv.i

conv.i2

conv.i4

conv.i8

conv.ovf.i2.un

conv.ovf.u2.un

conv.r4

conv.r8

conv.u

conv.u1

conv.u2

conv.u4

conv.u8

cpbk - 소스 주소에서 대상 주소로 복사합니다.

cupid - EFLAGS 레지스터의 ID 플래그 (비트 21)는 CPUID 명령에 대한 지원

cwde

daa - 압축 된 두 BCD 값의 합계를 조정

das - 압축 된 BCD 값 두 개를 뺀 결과를 조정

dec - 피연산자에서 1을 뺀

dup - 연속된 여러 값을 초기화

emms - x87 FPU 태그 단어에 있는 모든 태그의 값을 비워 둡니다

endfilter - 제어를 전달

Endfinally - 제어를 전달

enter - 프로 시저에 대한 스택 프레임 생성

fabs - 부호 비트를 지워 피연산자의 절대 값

fadd

faddp

fchs - ST (0)의 부호 비트를 변경.

Fclex - 예외 플래그 삭제

fiadd

fistp

fld1

fldcw - 16 비트 소스 피연산자를 FPU 제어 워드에 로드

fmulp

fnstcw - FPU 제어 워드 저장

fnstsw - FPU 상태 단어 저장

fsqrt - 제곱근을 계산하고 결과를 ST (0)에 저장

Fst - 부동 소수점 값 저장

Fstp - "

Fstsw - FPU 상태 단어 저장

fxch - 레지스터 ST (0) 및 ST (i)의 내용을 교환

hlt - 명령 실행을 중지하고 프로세서를 HALT 상태로 만듦

icebp

in - 두 번째 피연산자 (소스 피연산자)로 지정된 I / O 포트에서 대상 피연산자 (첫 번째 피연산자)로 값을 복사합니다.

inc - CF 플래그의 상태를 유지하면서 대상 피연산자에 1을 추가

Initobj - 0으로 초기화합니다.

Insb - 포트에서 문자열로 입력

Insd - "

Int - 인터럽트 발생

into

iret - 인터럽트 리턴

isinst - 객체 참조 (O 형)가 특정 클래스의 인스턴스인지 테스트합니다.

kmovw

lahf - 이 명령은 호환성 모드 및 레거시 모드에서 위에서 설명한대로 실행됩니다.

lea - 두 번째 피연산자 (소스 피연산자)의 유효 주소를 계산하여 첫 번째 피연산자 (대상 피연산자)에 저장합니다.

leave - 이전 ENTER 명령으로 설정된 스택 프레임 을 해제합니다.

leave.s

les

lfence - LFENCE 명령 이전에 발행 된 모든 메모리 에서로드 명령에 대해 직렬화 조작을 수행합니다.

here

Lodsb - 1byte를 읽어와서 저장, 배열 관련

Lodsd - 배열 관련

loop

loope

loopne

neg - 피연산자 값 (대상 피연산자)을 2의 보수로 바꿉니다.

Newarr - 0부터 시작하고 요소가 특정 형식인 새 1 차원 배열에 대한 개체 참조를 계산 스택으로 푸시 합니다.

Newobj - 개체 참조(0 형식)를 계산 스택으로 푸시 하여 값 형식의 새 개체나 새 인스턴스를 만듭니다.

nop - 이 명령은 작동하지 않습니다.

out - I / O 포트로 복사합니다.

Outsb - 지정된 IO 포트로 데이터 복사

Outsd - 문자열 출력

rcl - carry left

rcr - carry right

rethrow - 예외를 throw

rol - rotate left

ror - rotate right

sahf - AH를 플래그로 저장

sal - shift left

sar - shift right

scasb - 문자열 처리

scads - 문자열 처리

sfence - fence 저장

shl - shift left

shld - Double Precision Shift Left

shr - shift right

shr.un

shrd - Double Precision Shift Right

shufps - 단 정밀도 부동 소수점 값의 4 배의 팩형 인터리브 셔플

sizeof - 원소 개수 반환

starg.s

tail.

test - 논리적 비교

throw

ucomisd - 비 정렬 스칼라 배정 밀도 부동 소수점 값 비교 및 EFLAGS 설정

ucomiss - 비 정렬 스칼라 단 정밀도 부동 소수점 값 비교 및 EFLAGS 설정

ud2

unbox.any - 명령어에 지정된 유형의 박스 표시를 박스 형태로 변환

vbroadcastss

vextractf128

vfmadd213pd - 팩형 배정도 부동 소수점 값의 융

합 된 곱셈 추가

vfmadd213ps - 팩형 단 정밀도 부동 소수점의 융합
된 곱셈 추가

vinserftf128

volatile.

vorps

vpaddb

vpaddw

vpand

vpcmpgtd

vpor

vpshufd

vpsrlq

vpsubd

vpxor

vshufps

vsubpd

vsubps

vxorps

vzeroupper - YMM 레지스터의 상위 비트 제로

wait

xadd - 교환 및 추가

xchg - 레지스터가있는 교환 레지스터 / 메모리

xlat