

Access

Young W. Lim

2016-11-09 Wed

1 Introduction

- References
- IA32 Operand Forms

"Self-service Linux: Mastering the Art of Problem Determination", Mark Wilding
"Computer Architecture: A Programmer's Perspective", Bryant & O'Hallaron

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

Operand Types

- 1 Immediate Operand Type
- 2 Register Operand Type
- 3 Memory Reference Type

1) Immediate Operand Type

- constant values
- \$ followed by integer number
- only one or two bytes of 4 bytes integer

2) Register Operand Type

- denote the content of a register
 - 8 32-bit registers for double word operations
 - 8 8-bit registers for a byte operation
 - Ea : an arbitrary register a
 - $R[Ea]$ the value of an Ea register
 - view the set of registers as an array R
 - indexed by register identifiers

3) Memory Reference Type

- access some memory location
 - according to the computed address
 - effective address
- view the memory as a large array of bytes
- $Mb[Addr]$: the by b-byte value stored in memory starting at Addr
- addressing modes : allowing different forms of memory references
 - Imm : immediate offset
 - Eb : a base register
 - Ei : an index register
 - s : a scale factor (1, 2, 4, 8)

Addressing Modes

Imm	$M[\text{Imm}]$	Absolute
(Ea)	$M[R[\text{Ea}]]$	Indirect
Imm (Eb)	$M[\text{Imm} + R[\text{Eb}]]$	Base + displace
(Eb, Ei)	$M[R[\text{Eb}] + R[\text{Ei}]]$	Indexed
Imm (Eb, Ei)	$M[\text{Imm} + R[\text{Eb}] + R[\text{Ei}]]$	Indexed
(, Ei, s)	$M[R[\text{Ei}] * s]$	Scaled Indexed
Imm (, Ei, s)	$M[\text{Imm} + R[\text{Ei}] * s]$	Scaled Indexed
(Eb, Ei, s)	$M[R[\text{Eb}] + R[\text{Ei}] * s]$	Scaled Indexed
Imm (Eb, Ei, s)	$M[\text{Imm} + R[\text{Eb}] + R[\text{Ei}] * s]$	Scaled Indexed

- Imm : $M[\text{Imm}]$: Absolute
- (Ea) : $M[R[\text{Ea}]]$: Indirect
- Imm (Eb) : $M[\text{Imm} + R[\text{Eb}]]$: Base + displacement
- (Eb, Ei) : $M[R[\text{Eb}] + R[\text{Ei}]]$: Indexed
- Imm (Eb, Ei) : $M[\text{Imm} + R[\text{Eb}] + R[\text{Ei}]]$: Indexed
- (, Ei, s) : $M[R[\text{Ei}] * s]$: Scaled Indexed
- Imm (, Ei, s) : $M[\text{Imm} + R[\text{Ei}] * s]$: Scaled Indexed
- (Eb, Ei, s) : $M[R[\text{Eb}] + R[\text{Ei}] * s]$: Scaled Indexed
- Imm (Eb, Ei, s) : $M[\text{Imm} + R[\text{Eb}] + R[\text{Ei}] * s]$: Scaled Indexed