**Nama : Muhammad Yulianto**

**NIM : L200150057**

**Set Instruksi** didefinisikan sebagai suatu aspek dalam arsitektur komputer yang dapat dilihat oleh para pemrogram. Secara umum, ISA ini mencakup jenis data yang didukung, jenis instruksi yang dipakai, jenis register, mode pengalamatan, arsitektur memori, penanganan interupsi, eksepsi, dan operasi I/O eksternalnya (jika ada).

ISA merupakan sebuah spesifikasi dari kumpulan semua kode-kode biner (*opcode*) yang diimplementasikan dalam bentuk aslinya (*native form*) dalam sebuah desain prosesor tertentu. Kumpulan*opcode* tersebut, umumnya disebut sebagai **bahasa mesin** (*machine language*) untuk ISA yang bersangkutan. ISA yang populer digunakan adalah set instruksi untuk chip Intel x86, IA-64, IBM PowerPC, Motorola 68000, Sun SPARC, DEC Alpha, dan lain-lain.

ISA kadang-kadang digunakan untuk membedakan kumpulan karakteristik yang disebut di atas dengan mikroarsitektur prosesor, yang merupakan kumpulan teknik desain prosesor untuk mengimplementasikan set instruksi (mencakup *microcode*, *pipeline*, *sistem cache*, manajemen daya, dan lainnya). Komputer-komputer dengan mikroarsitektur berbeda dapat saling berbagi set instruksi yang sama. Sebagai contoh, prosesor Intel Pentium dan prosesor AMD Athlon mengimplementasikan versi yang hampir identik dari set instruksi Intel x86, tetapi jika ditinjau dari desain internalnya, perbedaannya sangat radikal. Konsep ini dapat diperluas untuk ISA-ISA yang unik seperti TIMI yang terdapat dalam IBM System/38 dan IBM IAS/400. TIMI merupakan sebuah ISA yang diimplementasikan sebagai perangkat lunak level rendah yang berfungsi sebagai mesin virtual. TIMI didesain untuk meningkatkan masa hidup sebuah platform dan aplikasi yang ditulis untuknya, sehingga mengizinkan platform tersebut agar dapat dipindahkan ke perangkat keras yang sama sekali berbeda tanpa harus memodifikasi perangkat lunak (kecuali yang berkaitan dengan TIMI). Hal ini membuat IBM dapat memindahkan platform AS/400 dari arsitektur mikroprosesor CISC ke arsitektur mikroprosesor POWER tanpa harus menulis ulang bagian-bagian dari dalam sistem operasi atau perangkat lunak yang diasosiasikan dengannya.

Ketika mendesain mikroarsitektur, para desainer menggunakan Register Transfer Language (RTL) untuk mendefinisikan operasi dari setiap instruksi yang terdapat dalam ISA.

Sebuah ISA juga dapat diemulasikan dalam bentuk perangkat lunak oleh sebuah interpreter. Karena terjadi translasi tambahan yang dibutuhkan untuk melakukan emulasi, hal ini memang menjadikannya lebih lambat jika dibandingkan dengan menjalankan program secara langsung di atas perangkat keras yang mengimplementasikan ISA tersebut. Akhir-akhir ini, banyak vendor ISA atau mikroarsitektur yang baru membuat perangkat lunak emulator yang dapat digunakan oleh para pengembang perangkat lunak sebelum implementasi dalam bentuk perangkat keras dirilis oleh vendor.

Daftar ISA di bawah ini tidak dapat dikatakan komprehensif, mengingat banyaknya arsitektur lama yang tidak digunakan lagi saat ini atau adanya ISA yang baru dibuat oleh para desainer

**Fitur Extensi Set Intruction Pada Prosesor Intel x86**

#### FPU ( x87 ) – Floating-point-unit ( FPU ) instructions

x87 adalah bagian dari x86 arsitektur set instruksi  floating point. Ini berasal sebagai perpanjangan dari 8086 set instruksi dalam bentuk opsional floating point coprocessor yang bekerja bersama-sama dengan CPU x86. microchip ini memiliki nama berakhiran "87". Ini juga dikenal sebagai NPX ( Numeric Processor eXstension ). Seperti ekstensi lain untuk set instruksi dasar, x87-petunjuk tidak ketat diperlukan untuk membangun program kerja, tetapi menyediakan hardware dan microcode implementasi tugas numerik umum, sehingga tugas-tugas ini akan dilakukan jauh lebih cepat daripada kode mesin biasa. Instruksi x87 termasuk instruksi untuk operasi dasar floating point seperti penambahan, pengurangan dan perbandingan, tetapi juga untuk operasi numerik yang lebih kompleks, seperti perhitungan tangen fungsi dan kebalikannya.

#### MMX

MMX ( Matrix Math eXtention ) digunakan untuk olah data matrix untuk extensi SIMD, diperkenalkan pada tahun 1997 dengan P5 berbasis Pentium mikroprosesor , ditunjuk sebagai "Pentium with MMX Technology". Dikembangkan dan diperkenalkan pertamakali pada Intel i860, dan sebelumnya i750 Intel prosesor pixel video. MMX adalah kemampuan tambahan prosesor yang didukung pada IA-32 prosesor Intel dan vendor lainnya.

#### MMX Extended

MMX ( Matrix Math eXtention ) seperti namanya ini merupakan perpanjangan dari MMX, termasuk dalam Intel Streaming SIMD Extensions sejumlah instruksi baru yang diperpanjan dari fungsi MMX. AMD dimasukkan bagian ini instruksi MMX diperpanjang ke Athlon mikroarsitektur. Instruksi ini umumnya dikenal sebagai "MMX Extended instruction". Instruksi ini memulai debutnya pada Mei 1999, di Intel Pentium III Processor, sebagai bagian dari SSE set instruksi. Pada akhir Juni 1999, AMD Athlon prosesor dirilis yang membawa instruksi MMX extended, tetapi tidak untuk fungsi SSE.

#### SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4, SSE4a

SSE ( Streamin SIMB Extention ) Pertama kali pada Intel IA-32, SIMD adalah MMX set instruksi. MMX memiliki dua masalah utama: itu digunakan kembali ada floating point register membuat CPU tidak dapat bekerja pada kedua floating point dan data SIMD pada saat yang sama, dan itu hanya bekerja pada bilangan bulat . SSE instruksi floating point beroperasi pada independen daftar set baru (register XMM), dan menambahkan beberapa instruksi integer yang bekerja pada register MMX.  
  
SSE kemudian diperluas oleh Intel untuk SSE2 , SSE3 , SSSE3 , dan SSE4 . Karena mendukung floating point matematika, itu memiliki aplikasi yang lebih luas daripada MMX dan menjadi lebih populer. Penambahan dukungan integer dalam SSE2 membuat MMX sebagian besar berlebihan, meskipun kinerja kenaikan lebih lanjut dapat dicapai dalam beberapa situasi dengan menggunakan MMX secara paralel dengan operasi SSE.

#### XSAVE

Melakukan penuh atau parsial hemat komponen prosesor ke daerah XSAVE yang terletak di alamat memori yang ditentukan oleh operan tujuan. Implisit EDX: EAX pasangan Register menentukan instruksi 64-bit. Komponen negara tertentu disimpan sesuai dengan bit diatur dalam meminta-fitur bitmap ( RFBM ), yang memerlukan logika-AND dari EDX: EAX dan XCR0.

#### AVX

AVX ( Advanced Vector Extensions ) adalah ekstensi untuk x86 arsitektur untuk mikroprosesor dari Intel dan AMD yang diusulkan oleh Intel pada Maret 2008 dan pertama didukung oleh Intel dengan Sandy Bridge pengiriman prosesor di Quartal pertama ( 1 Quartal = 4 bulan ) 2011 dan kemudian oleh AMD dengan Bulldozer pengiriman prosesor di Q3 2011. AVX menyediakan fitur baru, instruksi baru dan skema pengkodean baru.

#### AVX2

AVX2 memperluas paling bulat perintah untuk 256 bit dan memperkenalkan FMA . AVX-512 memperluas AVX dukungan 512-bit memanfaatkan EVEX awalan encoding yang diusulkan oleh Intel pada bulan Juli tahun 2013 dan pertama didukung oleh Intel dengan Knights Landing prosesor dijadwalkan untuk pada tahun 2015.

#### FMA

FMA ( Fused Multiply Add ) seperti namanya, intruksi ini digunakan untuk melakukan multi intruksi dan perluasan ke 128 dan 256-bit Streaming SIMD Extensions yang ada di x86 mikroprosesor set instruksi.

Set Instruksi ARM(Android)

ARM dan set instruksi Thumb dapat secara luas diklasifikasikan ke dalam kelompok-kelompok fungsional berikut:  
1. Instruksi percabangan dan Pengendalian: Petunjuk seperti panggilan subrutin, perulangan dan mengubah negara antara ARM dan jatuh Thumb bawah kategori instruksi.  
2. Daftar instruksi Load dan Store: Memuat nilai-nilai dari register tunggal untuk dan dari memori yang tercakup dalam jenis instruksi. Nilai-nilai mungkin 32 bit kata, 16-bit kata atau setengah nilai unsigned 8 bit.  
3. Beberapa Daftar Instruksi Load dan Toko: Memfasilitasi gerakan ke sana kemari antara isi dari register ganda, digunakan dalam operasi blok dan operasi stack.  
4. Petunjuk Pengolahan Data: Operasi seperti penambahan, pengurangan atau bitwise logika pada isi dari register yang dilakukan oleh jenis instruksi.  
5. Status Daftar Instruksi akses: Instruksi ini terutama memindahkan isi antara register status dan GPR.  
6. Coprocessor Petunjuk: ini memberikan kerangka umum untuk memperluas arsitektur ARM.  
Siapapun dengan pengetahuan sebelumnya dari arsitektur mikroprosesor dasar akan mengenali kemiripan antara berbagai klasifikasi dan juga set instruksi. Namun ada beberapa fitur yang tidak dapat disesuaikan oleh pengguna dirinya dalam ARMS dan pekerjaan yang tersisa untuk compiler itu sendiri. Dengan evolusi mesin Cortex, prosesor sekarang telah dibagi menjadi 3 profil berdasarkan jenis aplikasi mereka menangani:  
1. Profil aplikasi: Ini adalah aplikasi prosesor khusus seperti Cortex-A8 yang fitur Memori dukungan manajemen (MMU) dan kinerja tinggi pada daya rendah.  
2. Real-time Profil: Dibuat untuk prosesor real time seperti Cortex-R4 yang memiliki memori yang dilindungi (MPU), dan latency rendah diperlukan untuk aplikasi real time.  
3. Profil Mikrokontroler: Perangkat ini dimaksudkan untuk perangkat mobile seperti Cortex-M3. Perilaku diprediksi merupakan prioritas utama dengan menghitung gerbang rendah dan menemukan digunakan dalam aplikasi tertentu tertanam.  
Dimulai dengan ARMS lama bekerja pada kecepatan siklus lambat, mesin-mesin telah akhirnya berkembang menjadi mesin kinerja tinggi menawarkan kinerja baterai yang lebih baik dan konsumsi daya yang lebih kecil.