

PERTEMUAN 15

SUPERSCALAR

1. Pengertian

Superscalar adalah sebuah unitprocessor yang dapat mengeksekusi dua atau lebih operasi scalar dalam bentuk paralel. Superscalar merupakan salah satu rancangan untuk meningkatkan kecepatan CPU.

Kebanyakan dari komputer saat ini menggunakan mekanisme superscalar ini. Standar pipeline yang digunakan adalah untuk pengolahan bilangan matematika integer (bilangan bulat, bilangan yang tidak memiliki pecahan), kebanyakan CPU juga memiliki kemampuan untuk pengolahan untuk data floating point (bilangan berkoma). Pipeline yang mengolah integer dapat juga digunakan untuk mengolah data bertipe floating point ini, namun untuk aplikasi tertentu, terutama untuk aplikasi keperluan ilmiah CPU yang memiliki kemampuan pengolahan floating point dapat meningkatkan kecepatan prosesnya secara dramatis.

Peristiwa menarik yang bisa dilakukan dengan metoda superscalar ini adalah dalam hal memperkirakan pencabangan instruksi (branch prediction) serta perkiraan eksekusi perintah (speculative execution). Peristiwa ini sangat menguntungkan buat program yang membutuhkan pencabangan dari kelompok instruksi yang dijalankannya.

Program yang terdiri dari kelompok perintah bercabang ini sering digunakan dalam pemrograman. Contohnya dalam menentukan aktifitas yang dilakukan oleh suatu sistem berdasarkan umur seseorang yang sedang diolahnya, katakanlah jika umur yang bersangkutan lebih dari 18 tahun, maka akan diberlakukan instruksi yang berhubungan dengan umur tersebut, anggaplah seseorang tersebut dianggap telah dewasa, sedangkan untuk kondisi lainnya dianggap belum dewasa. Tentu perlakuannya akan dibedakan sesuai dengan sistem yang sedang dijalankan.

Lalu apa yang dilakukan oleh CPU untuk hal ini? Komputer akan membandingkan nilai umur data yang diperolehnya dengan 18 tahun sehingga komputer dapat menentukan langkah dan sikap yang harus diambilnya berdasarkan hasil perbandingan tersebut. Sikap yang diambil tentu akan diambil berdasarkan pencabangan yang ada.

Pada CPU yang mendukung perintah pencabangan ini, CPU membutuhkan lumayan banyak clock cycle, mengingat CPU menempatkan semuanya pada pipeline dan menemukan perintah berikutnya yang akan dieksekusinya. Sirkuit untuk branch prediction melakukan pekerjaan ini bekerja sama dengan pipeline, yang dilakukan sebelum proses di ALU dilaksanakan, dan memperkirakan hasil dari pencabangan tersebut.

Jika CPU berfikir bahwa branch akan menuju suatu cabang, biasanya berdasarkan pekerjaan sebelumnya, maka perintah berikutnya sudah dipersiapkan untuk dieksekusi berikut data-datanya, bahkan dengan adanya pipeline ini, bila tidak diperlukan suatu referensi dari instruksi terakhir, maka bisa dilaksanakan dengan segera, karena data dan instruksi yang dibutuhkan telah dipersiapkan sebelumnya.

Dalam hal speculative execution, artinya CPU akan menggunakan melakukan perhitungan pada pipeline yang berbeda berdasarkan kemungkinan yang diperkirakan oleh komputer. Jika kemungkinan yang dilakukan oleh komputer tepat, maka hasilnya sudah bisa diambil langsung dan tinggal melanjutkan perintah berikutnya, sedangkan jika kemungkinan yang diperkirakan oleh komputer tidak tepat, maka akan dilaksanakan kemungkinan lain sesuai dengan logika instruksi tersebut.

Teknik yang digunakan untuk pipeline dan superscalar ini bisa melaksanakan branch prediction dan speculative execution tentunya membutuhkan ekstra transistor yang tidak sedikit untuk hal tersebut.

Sebagai perbandingan, komputer yang membangkitkan pemrosesan pada PC pertama yang dikeluarkan oleh IBM pada mesin 8088 memiliki sekitar 29.000 transistor. Sedangkan pada mesin Pentium III, dengan teknologi superscalar dan superpipeline, mendukung branch prediction, speculative execution serta berbagai kemampuan lainnya memiliki sekitar 7,5 juta transistor. Beberapa CPU terkini lainnya seperti HP 8500 memiliki sekitar 140 juta transistor.

Superscalar ini mampu menjalankan Instruction Level Parallelism dengan satu prosesor. Superscalar dapat diaplikasikan di RISC dan CISC, tapi pada umumnya RISC.

2. Alasan desain Superscalar

Sebagian besar operasi menggunakan besaran/nilai skalar Operasi ini memungkinkan peningkatan kinerja sistem hingga level tertentu Superscalar Implementation.

Proses fetch dari beberapa instruksi secara bersamaan. Logika untuk menentukan ketergantungan sebenarnya yang meliputi nilai register Mekanisme untuk mengkomunikasikan nilai tersebut. Mekanisme untuk menginisialisasi instruksi paralel. Tersedianya sumber untuk eksekusi paralel dari beberapa instruksi. Mekanisme processing instruksi dengan urutan yg sesuai.

3. Contoh Instruksi

Add R1, R2, R3

Misalkan, instruksi

yang akan menambahkan isi register R1 dan R2 dan menempatkan jumlahnya dalam register R3. Isi dari register R1 dan R2 mula-mula akan ditransfer ke Unit aritmetika dan logika. Setelah operasi penambahan dilakukan, hasil penjumlahan tersebut akan ditransfer ke register R3. Prosesor dapat membaca instruksi selanjutnya dari memori, sementara operasi penambahan dilakukan. Kemudian jika instruksi tersebut juga menggunakan ALU, operand-nya dapat ditransfer ke input ALU pada waktu yang sama dengan hasil instruksi **Add** ditransfer ke register R3.

Pada kasus ideal, jika semua instruksi ditumpuk ke derajat yang maksimum yang mungkin dilakukan, maka eksekusi dilanjutkan pada kecepatan penyelesaian satu instruksi dalam tiap siklus detak prosesor. Instruksi individual, mungkin masih memerlukan beberapa siklus detak agar selesai dilakukan. Tetapi untuk tujuan perhitungan, prosesor superskalar umumnya mampu melakukannya dalam tiap siklus.

Prosesor superskalar umumnya menggunakan beberapa unit fungsional, menciptakan jalur paralel di mana berbagai instruksi yang berbeda dapat dieksekusi secara paralel. Dengan pengaturan tersebut, maka dimungkinkan untuk memulai eksekusi beberapa instruksi secara paralel tiap siklus detak. Tentu saja, eksekusi paralel harus mempertahankan kebenaran logika program, sehingga hasil yang diperoleh harus sama dengan hasil dari eksekusi secara serial.

4. Contoh CPU yang menerapkan arsitektur superscalar :

- 486, Pentium, Pentium Pro, keluarga Intel Pentium, Intel Pentium Pro, Intel Pentium II, Intel Pentium III, Intel Itanium, Intel Xeon, Intel Pentium 4, Intel Pentium M, Intel Core dari Intel Corporation; keluarga AMD K5, AMD K6, AMD Athlon, AMD Athlon 64, dan AMD Opteron

Superscalar Processor Design

- Use PowerPC 604 as case study
- Speculative Execution, Register Renaming, Branch Prediction

More Superscalar Examples

- MIPS R10000

- DEC Alpha 21264

5. Implementasi Superscalar

1. Proses fetch dari beberapa instruksi secara bersamaan.
2. Logika untuk menentukan ketergantungan sebenarnya yang meliputi nilai register.
3. Mekanisme untuk mengkomunikasikan nilai tersebut.
4. Mekanisme untuk menginisialisasi instruksi paralel.
5. Tersedianya sumber untuk eksekusi paralel dari beberapa instruksi.
6. Mekanisme processing instruksi dengan urutan yg sesuai.

Pada penjelasan diatas bias diterangkan bahwa untuk superscalar dapat digunakan untuk berbagai keperluan dan dapat diimplementasikan pada perangkat prosessor seperti :

Pentium 4

Pada Pentium, implementasi superscalar dapat dijabarkan sebagai berikut :

- 80486 – CISC.
- Pentium.
- ada beberapa komponen superscalar.
- 2 unit eksekusi integer yang terpisah.
- Pentium Pro – Full superscalar.
- Memperhalus models subsequent & Meningkatkan design superscalar.

6. Permasalahan pada Superscalar

Kemampuan dari computer superscalar dapat dilihat dari banyaknya instruksi yang dapat diproses secara parallel. Superscalar hingga derajat tertentu dapat terjadi jika computer mampu mengatasi permasalahan ketergantungan yang ada pada suatu instruksi, diantaranya adalah : ketergantungan data, ketergantungan procedural, ketergantungan sumber unit, ketergantungan output dan antiketergantungan. Dari lima ketergantungan tersebut, sampai sekarang belum ada computer yang mampu mengatasinya secara total. Bagaimanapun computer akan menemukan beberapa ketergantungan dalam menyelesaikan suatu program.

Dari rancangan computer yang ada, hanya mampu meminimalisasikan beberapa ketergantungan saja, walaupun demikian ternyata dengan usaha ini mampu diciptakan computer dengan kinerja yang jauh lebih baik dari generasi-generasi sebelumnya. Adapun usaha yang dilakukan dalam peningkatan kinerja superscalar tersebut diantaranya : desain pipeline (*out of order issue and out of order completion*), memperbanyak tahapan pada pipeline, penduplikasian sumber unit (contoh : unit fungsional /ALU, memori, bus, cache, dll), desain micro operation (pada system CICS), desain unit prediksi percabangan, dan renaming allocation. Kebanyakan computer lebih menekankan pada upaya penduplikasian

unit fungsional baik integer maupun floating point. Untuk itu pada pembahasan ini akan kita bahas salah satu upaya peningkatan kinerja prosesor superscalar secara khusus dalam hal rekonfigurasi *floating point unit* (FPU).