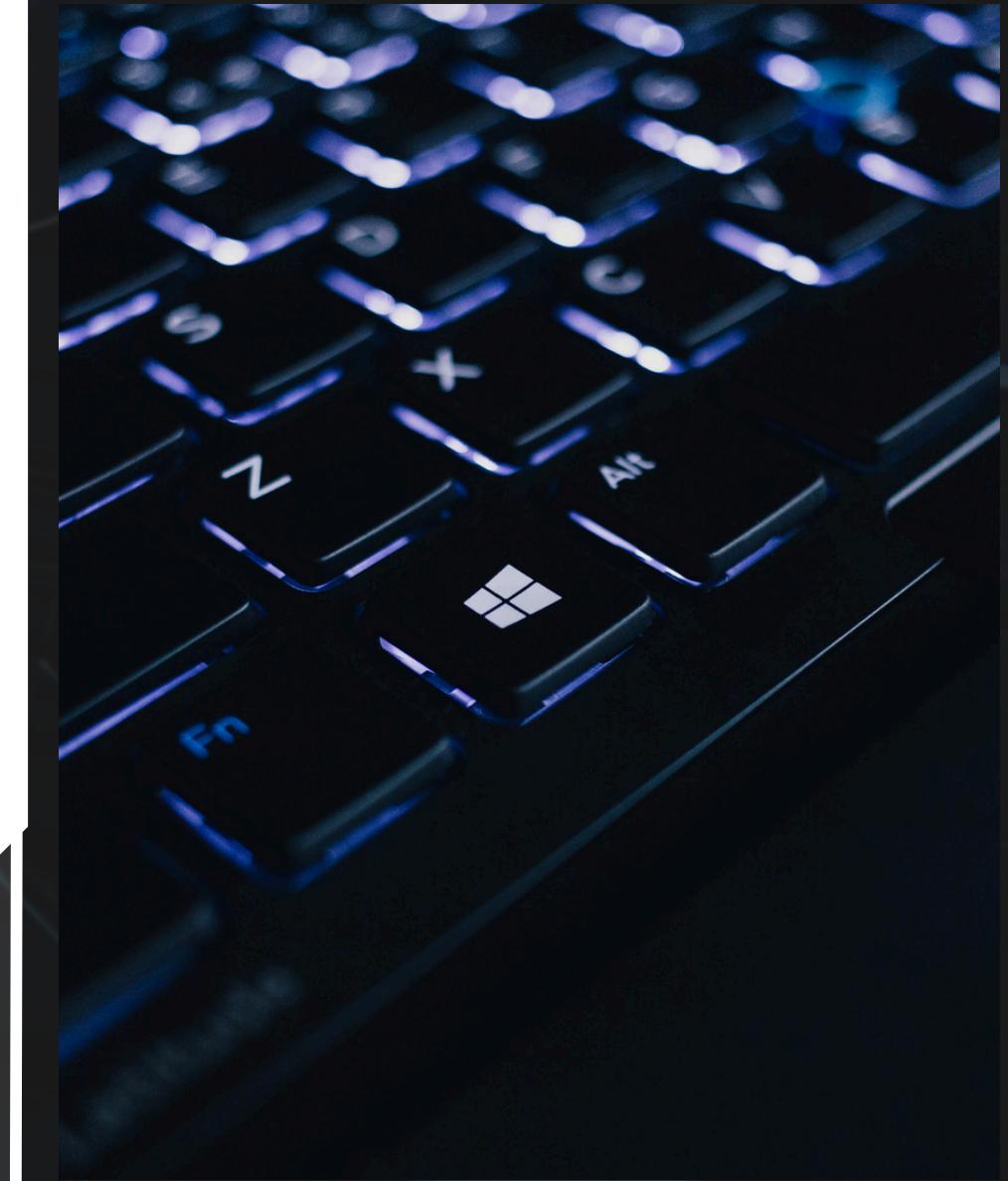
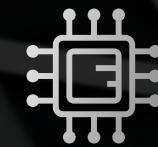


GROUP 18

COMPUTER SUPERSCALAR

Final Project - Perancangan Sistem Digital





GROUP 18

OUR TEAM



Khalisa Zahra Maulana
2406425395



Otniel Kristian Sianturi
2406401571



Putri Ayu Pembayun M
2406422304



Salsabila Maharani Mumtaz
2406348156



GROUP 18

ABOUT COMPUTERS SUPERSCALAR

Untuk memberikan sedikit konteks, prosesor modern bekerja dengan mencoba mengeksekusi instruksi sebanyak mungkin dalam waktu sesingkat mungkin. Salah satu caranya adalah dengan menjalankan dua instruksi sekaligus. Tapi tentu saja, tidak semua instruksi aman dijalankan bareng-bareng; ada instruksi yang saling bergantung atau memakai resource yang sama.

Nah, tugas proyek kami adalah membuat sistem yang bisa mengambil keputusan:

“Instruksi ini aman dijalankan berdua, atau harus jalan satu per satu?”

Di sini kami belajar bagaimana mengenali konflik, bagaimana menganalisis aliran data antar instruksi, dan bagaimana prosesor membuat keputusan yang terlihat “pintar”.

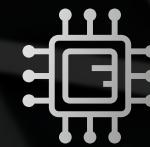




PROJECT DESCRIPTION

Secara sederhana, sistem yang kami buat bekerja seperti “penyaring”. Setiap kali dua instruksi datang, sistem ini membaca keduanya, memeriksa apakah ada potensi tabrakan data atau tabrakan penggunaan resource, dan menentukan apakah dua instruksi itu bisa berjalan paralel atau tidak.

Agar proses ini berjalan rapi, kami membangun beberapa bagian seperti decoder, pendekripsi hazard, pembuat sinyal kontrol, pengecek branch, dan logic yang menentukan keputusan akhirnya. Semua bagian ini bekerja bersama, memastikan eksekusi instruksi tetap benar tanpa kehilangan kesempatan untuk mempercepat proses.

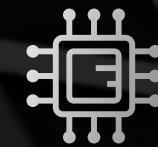


PROJECT IMPLEMENTATION

Component Instruction Decoder digunakan untuk memetakan input opcode 4-bit menjadi record dengan 8 flag boolean. Terdapat 2 komponen, karena implementasi komputer superscalar. Kemudian, menghindari conflict terdapat **Hazard Detection** untuk mendeteksi RAW, WAW, WAR, Structural, dan Control Hazard. Komponen **Control Signal Generator** berfungsi untuk mengonversi opcode menjadi sinyal kontrol 14-bit yang mengatur pin-pin. Pola bit yang dihasilkan komponen ini pada dasarnya menentukan jalur data mana saja yang harus diaktifkan untuk setiap jenis operasi, sebelum akhirnya divalidasi oleh Issue Logic.

Komponen **Branch Resolution Unit** bekerja secara kombinasional untuk mengevaluasi logika percabangan bersyarat (conditional branching) dengan cara membandingkan tipe instruksi branch yang sedang diproses dengan kondisi flag prosesor saat ini, yaitu zero_flag dan carry_flag. dan yang terakhir **Issue Logic** merupakan proses sekuensial yang disinkronkan dengan clock (clocked process) yang berfungsi mengintegrasikan semua hasil analisis dari komponen-komponen sebelumnya untuk kemudian membuat keputusan final eksekusi di setiap clock cycle.





PROJECT OBJECTIVES



Tujuan Utama

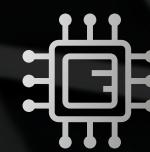
Proyek ini bertujuan memahami bagaimana sebuah prosesor menentukan apakah dua instruksi dapat dijalankan secara bersamaan tanpa konflik. Fokus kami bukan hanya membuat modul yang “berjalan”, tetapi memastikan sistem mampu mendeteksi hazard, mengatur sinyal kontrol yang benar, dan mereplikasi perilaku eksekusi prosesor secara realistik.



Optimalisasi Eksekusi Instruksi

Kami ingin mengevaluasi sejauh mana eksekusi paralel dapat meningkatkan performa ketika instruksi tidak saling bergantung. Dengan membandingkan jalur paralel dan sequential, proyek ini memberikan gambaran nyata tentang kapan parallelism benar-benar menguntungkan dan kapan tidak.





PROJECT TESTING

Metode testing prosesor menggunakan Simulation-Based Testing, Functional Testing, dan Hazard Detection Testing untuk menguji komponen utama melalui empat program berbeda. Program pertama menguji RAW hazard pada load-add sequence (9 cycles), program kedua menguji WAR dan WAW hazard pada store-move operations (9 cycles), program ketiga mendemonstrasikan 2-way parallel execution yang mengurangi siklus dari 14 menjadi 12 cycles (speedup 1.17x), dan program keempat menguji complex dependency pattern pada memory operations (13 cycles). Hasil pengujian membuktikan prosesor mampu mendeteksi berbagai hazard secara akurat, menangani dependency dengan tepat, mengidentifikasi peluang parallel execution, dan menjaga correctness komputasi.





CONCLUSION

- Proyek ini berhasil membuat sistem superscalar 2-way yang menentukan apakah dua instruksi bisa dieksekusi paralel atau harus sequential.
- Sistem memeriksa dependency dan hazard (RAW, WAR, WAW, structural, control) untuk mencegah konflik selama eksekusi.
- Modul-modul utama seperti Decoder, Hazard Detection, Control Signal Generator, Branch Resolution, dan Issue Logic bekerja bersama untuk menjaga eksekusi tetap benar.
- Hasil simulasi menunjukkan bahwa parallel execution hanya efektif ketika instruksi tidak saling bergantung.
- Proyek ini menunjukkan bagaimana keputusan eksekusi dibuat berdasarkan analisis dependency dan hazard secara langsung.





GROUP 18



THANK YOU!