A 32–bit Pipeline RISC CPU

from Verilog Code to Gata–Level Synthesis

王麒瑋

DISP

December 20, 2013

Outline

Outline

Roughly Speaking...

A CPU needs:

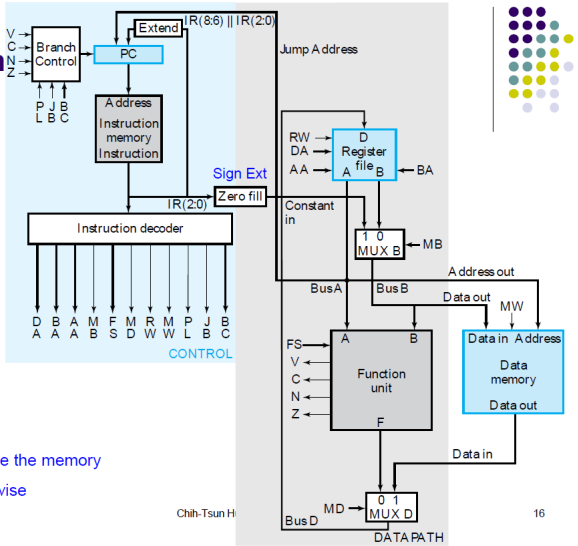
Registers: Memory Hierachy 的最上層 Storage Unit。CPU 可以直接讀取、寫入。

Instruction: CPU 聽得懂的話。以 $A + B = C$ 爲例，必須告訴 CPUA, B, C 的位置和它必須執行的運算 $'+'$。

ALU: Function Unit，執行各種運算的地方。

Cache: 比 Registers 慢一點，但容量較大的 Memory(SRAM)。現在的做法會將存 innstruction 和 data 的 cache 分開。

Block Diagram

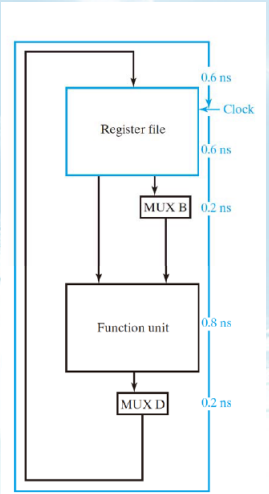


MW = 1: to write the memory

MW = 0: otherwise

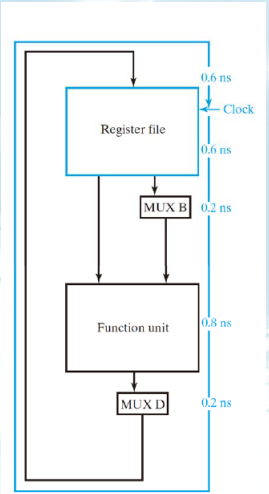
Clocks...

- 每個 Instruction 都需要數個 Clocks 來完成，包括從 Cache 讀入 Data、Instruction 到 Registers。



Clocks...

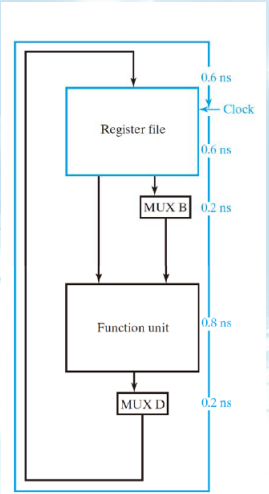
- 每個 Instruction 都需要數個 Clocks 來完成，包括從 Cache 讀入 Data、Instruction 到 Registers。
- How to increase efficiency ?



Outline – CPU Structure

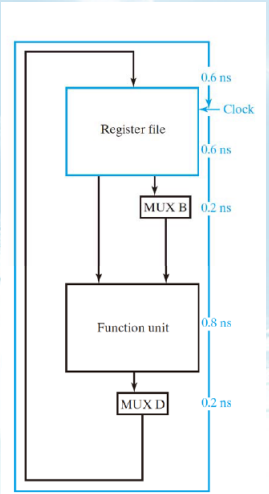
閒置村民...

- 從 Cache 讀資料到 Registers 時，Function Unit 閒置。



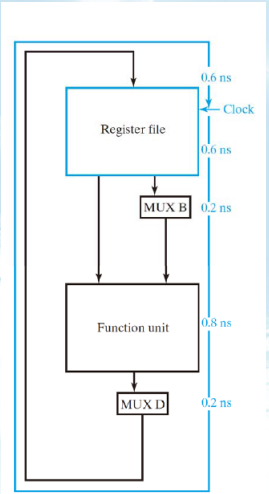
閒置村民...

- 從 Cache 讀資料到 Registers 時，Function Unit 閒置。
- Function Unit 運作時，Register file 閒置。



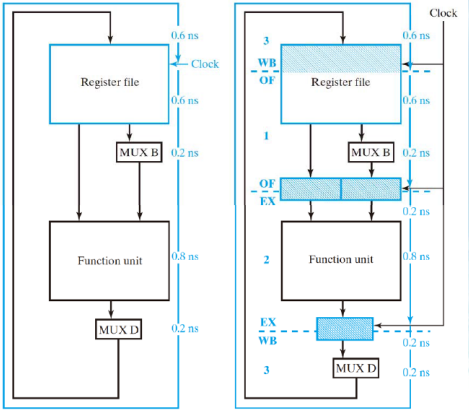
閒置村民...

- 從 Cache 讀資料到 Registers 時，Function Unit 閒置。
- Function Unit 運作時，Register file 閒置。
- 一起工作，不要偷懶。

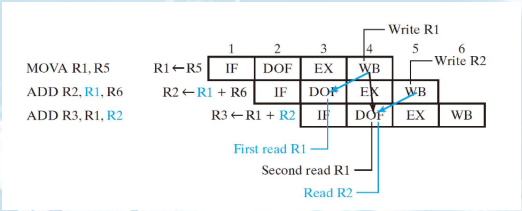


閒置村民...

- Single cycle:
  - Latency: 2.4ns
  - Max frequency:
    - 416.7MHz
- Pipeline:
  - Latency of a single operations: 2.8ns
  - Max delay between *pipeline platforms:* 1.0ns
  - Max frequency:
    - 1GHz

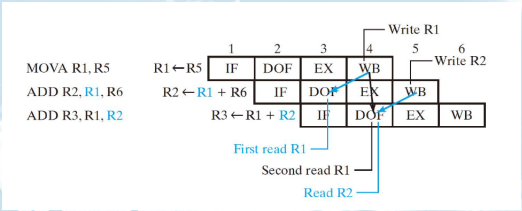


Hazards



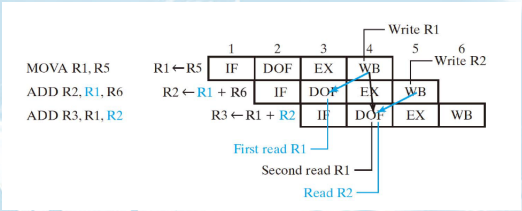
- However, other problems occur. It's called Data Hazards.

Hazards



- However, other problems occur. It's called Data Hazards.
- We don't go details but there're some methods to handle...

Hazards

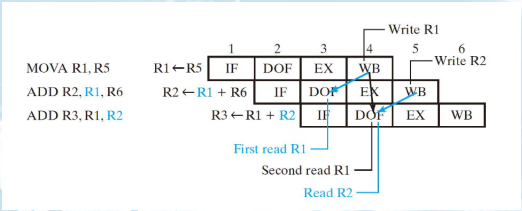


- However, other problems occur. It's called Data Hazards.
- We don't go details but there're some methods to handle...

Data Stall

Add 'bubble' to certain cycle.

Hazards



- However, other problems occur. It's called Data Hazards.
- We don't go details but there're some methods to handle...

Data Stall

Add 'bubble' to certain cycle.

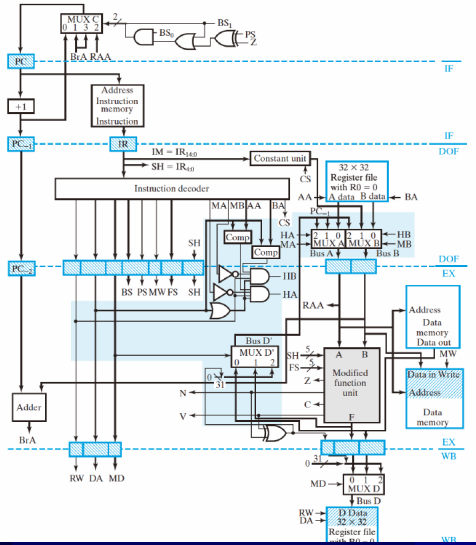
Data Forwarding

Don't wait for WB stage, just throw the computed–output to following instructions.

Pipeline CPU Structure

Pipeline RISC with Data Forwarding

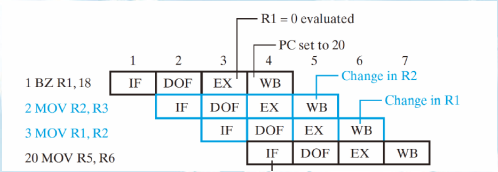
- An additional MUX D' for forwarding



Spring 2013 CS4125

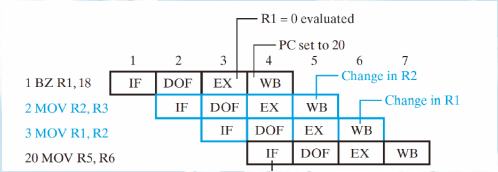
Branch–Hazard, or called Control Hazard

- After pipelined, the branch instruction has something wrong.



Branch–Hazard, or called Control Hazard

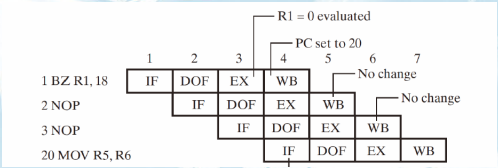
- After pipelined, the branch instruction has something wrong.



- The same strategy as Data Hazard can be taken.

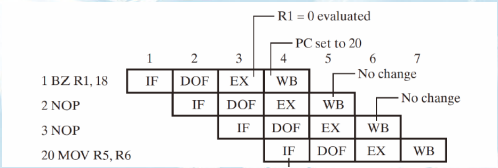
Adding NOPs

- just like Data Stall.



Adding NOPs

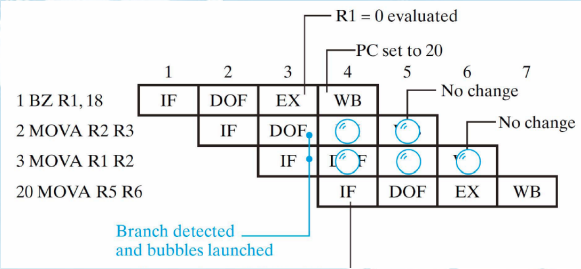
- just like Data Stall.



- Don't do anything before confirming branch targets.

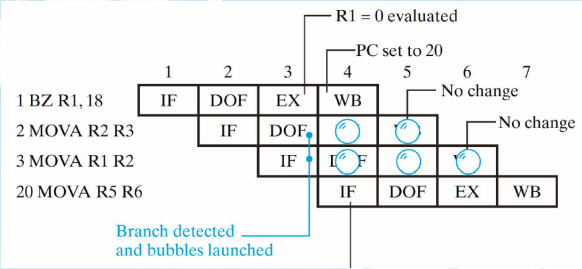
Branch Prediction

- However, We can guess what will happen.



Branch Prediction

- However, We can guess what will happen.

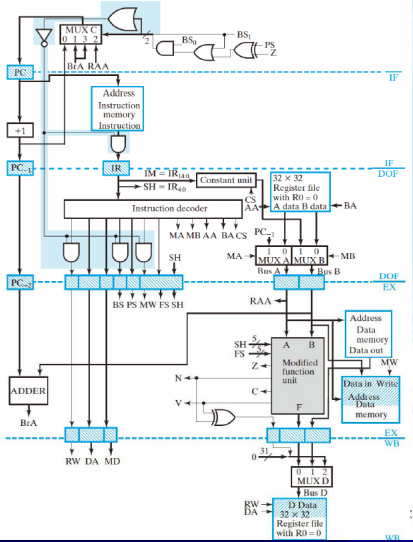


- And undo instructions if our guess is wrong.

Now the CPU looks like...

Pipeline RISC with Branch Prediction

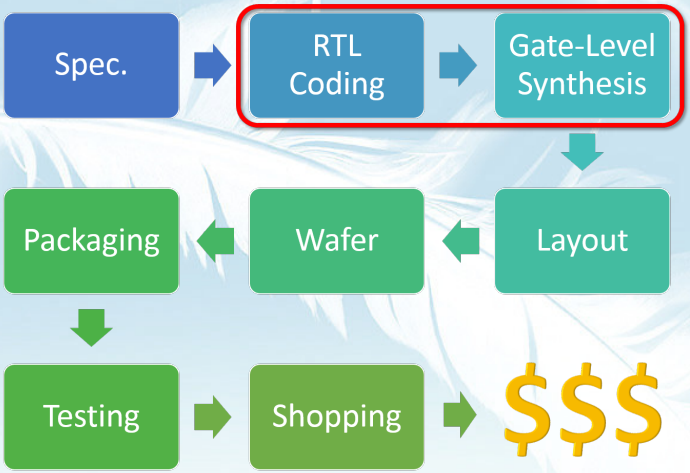
- Branch/Jump is controlled by MUX C
  - When the selection is not zero, branch or jump occurs
- A big OR is used to clear
  - RW and MW from writing data
  - BS to stop next branch
  - IR to replace the next instruction with a NOP



Spring 2013 CS4125

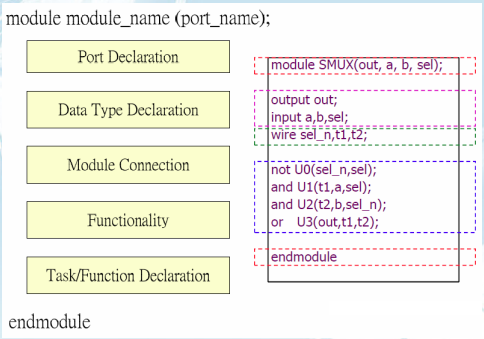
Outline

Digital IC Design Flow



RTL Coding – Verilog

- A kind of hardware description language(HDL).
- Module–Base



- alu.v
- register_file.v
- instruction_decorder.v
- risc.v

Outline – Implementation

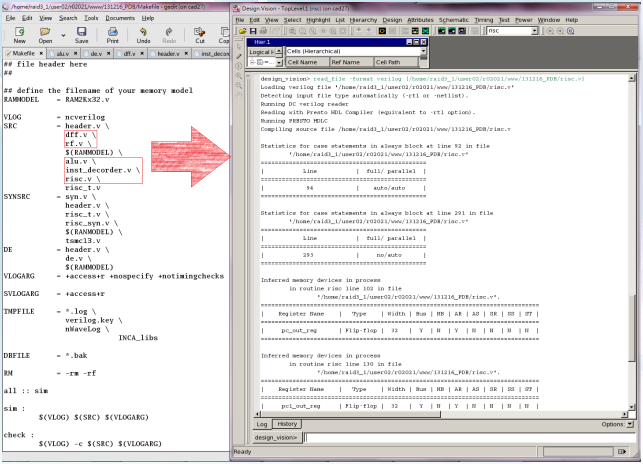
2. Implementation
   - IC Design Flow
   - RTL Coding
   - Gate–Level Synthesis

Tool Chain(EDA)

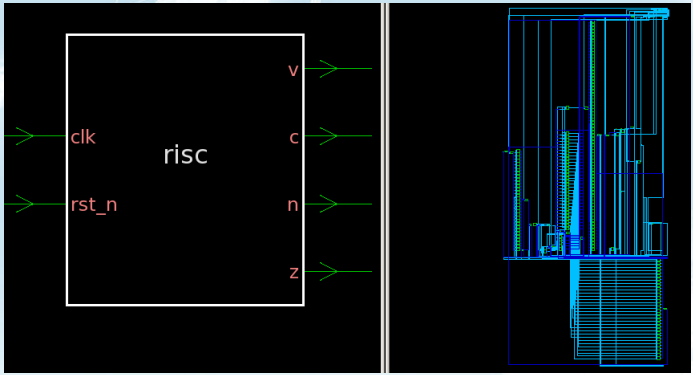
- There is a tool chain from spec. to alerttesting
- Here only introduce *Design Compiler*, used to synthesize RTL code to gate–level.
- *Virtusos→Avant!→Synopsys*

Reading Design

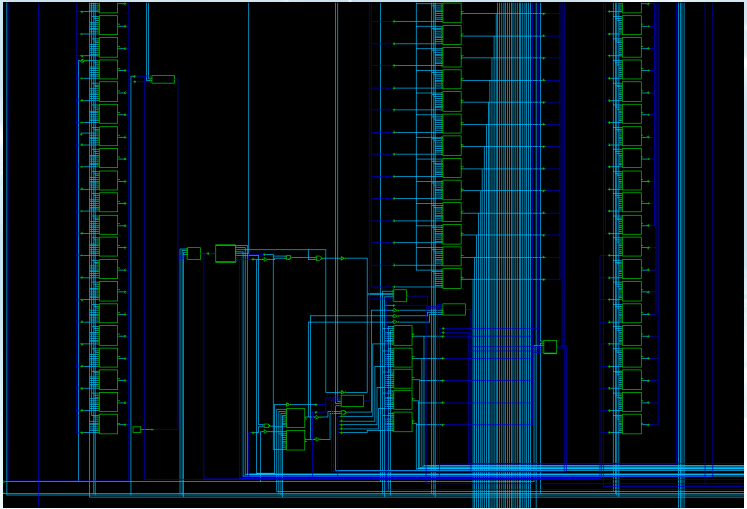
- 夯不啷噹地讀進所有.v 檔



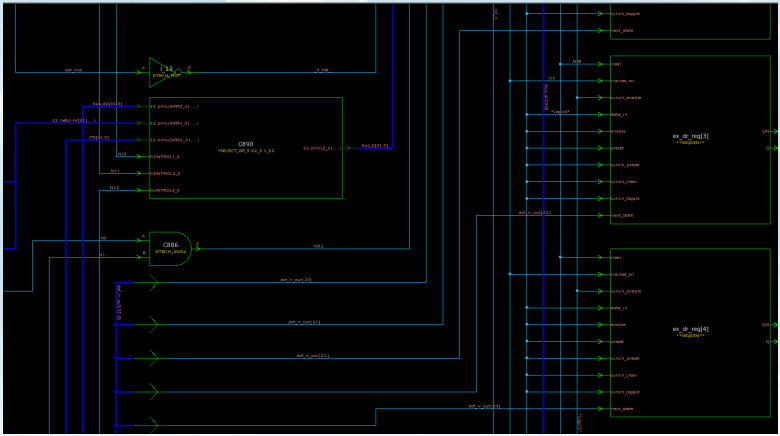
Schematic & Symbol – Top Module



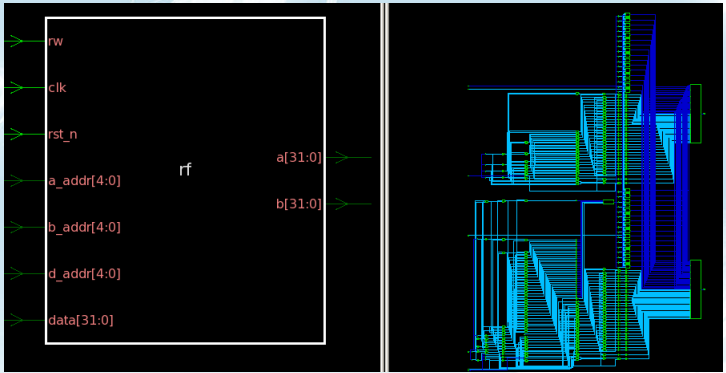
Schematic & Symbol – Top Module



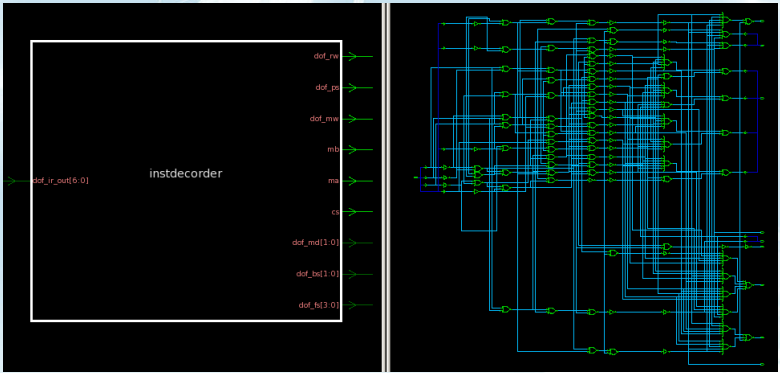
Schematic & Symbol – Top Module



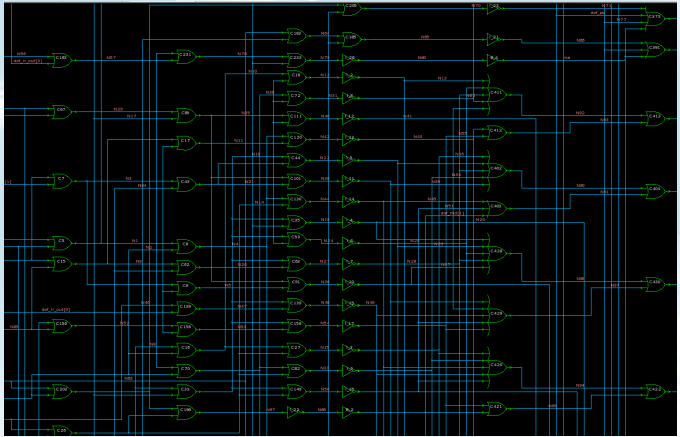
Schematic & Symbol – Register File



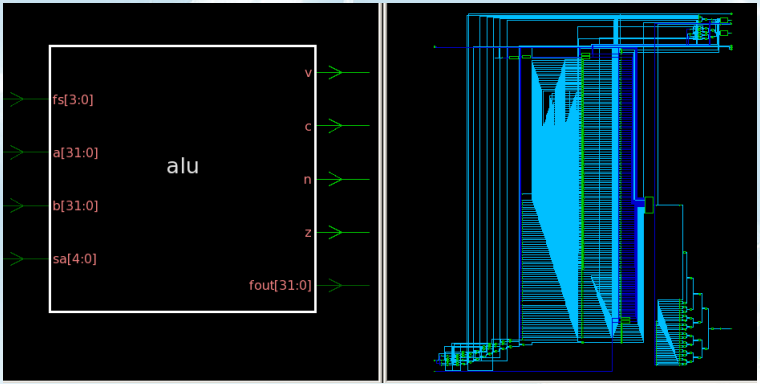
Schematic & Symbol – Instruction Decoder



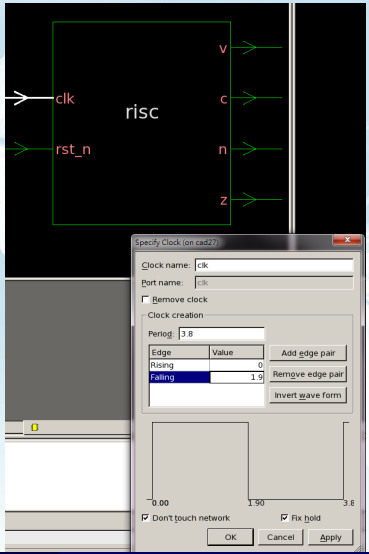
Schematic & Symbol – Instruction Decoder



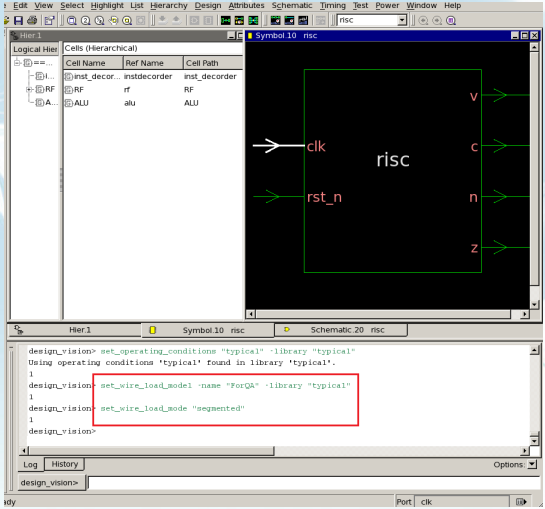
Schematic & Symbol – ALU



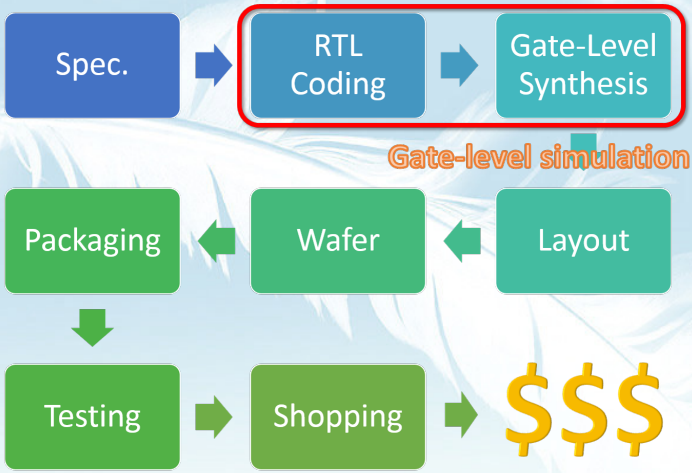
Compiling Design – Clock & Wire Load Model



Compiling Design – Clock & Wire Load Model



Gate–Level Simulation



Outline

Awful Programming Experience on the self–designing CPU...

- 世界上有 10 種人，懂二進位和不懂二進位的:

tt_bin.dat

```
@0 //This is a program calculating 1+2+3+.......n
0100010_10000_00000__000010000000000  //   main:  ADI  R16, R0, #1024
0010000_00010_10000__000000000000000  //          LD   R2, M[R16]
0000010_00001_00001_00010_0000000000  //   add:   ADD  R1, R1, R2
0100101_00010_00010__0000000000000001 //          SBI  R2, R2, #1
1001000_00000_00010__111111111111100  //          BNZ  R2, add (-4)
0100010_10010_00000__000010000000010  //   done:  ADI  R18, R0, #1026
0100000_00000_10010_00001_0000000000  //          ST   M[R18], R1
```

Awful Programming Experience on the self–designing CPU...

- 世界上有 10 種人，懂二進位和不懂二進位的:

tt_bin.dat

```
@0 //This is a program calculating 1+2+3+.......n
0100010_10000_00000__000010000000000  //   main:  ADI  R16, R0, #1024
0010000_00010_10000__000000000000000  //          LD   R2, M[R16]
0000010_00001_00001_00010_0000000000  //   add:   ADD  R1, R1, R2
0100101_00010_00010__0000000000000001 //          SBI  R2, R2, #1
1001000_00000_00010__111111111111100  //          BNZ  R2, add (-4)
0100010_10010_00000__000010000000010  //   done:  ADI  R18, R0, #1026
0100000_00000_10010_00001_0000000000  //          ST   M[R18], R1
```

- 完全依賴於指令集 (spec.) 的設定，所以現在新的 CPU 品牌很難普及。PC 至今還在使用 CISC 的原因也大約如此。

Awful Programming Experience on the self–designing CPU...

- 世界上有 10 種人，懂二進位和不懂二進位的:

tt_bin.dat

```
@0 //This is a program calculating 1+2+3+.......n
0100010_10000_00000__0000010000000000  //   main:  ADI  R16, R0, #1024
0010000_00010_10000__0000000000000000  //          LD   R2, M[R16]
0000010_00001_00001_00010_0000000000   //   add:   ADD  R1, R1, R2
0100101_00010_00010__0000000000000001  //          SBI  R2, R2, #1
1001000_00000_00010__1111111111111100  //          BNZ  R2, add (-4)
0100010_10010_00000__0000100000000010  //   done:  ADI  R18, R0, #1026
0100000_00000_10010_00001_0000000000   //          ST   M[R18], R1
```

- 完全依賴於指令集 (spec.) 的設定，所以現在新的 CPU 品牌很難普及。PC 至今還在使用 CISC 的原因也大約如此。
- ARM CPU 架構是 RISC。

Awful Programming Experience on the self–designing CPU...

- 世界上有 10 種人，懂二進位和不懂二進位的:

tt_bin.dat

```
@0 //This is a program calculating 1+2+3+.......n
0100010_10000_00000__0000010000000000  //   main:  ADI  R16, R0, #1024
0010000_00010_10000__0000000000000000  //          LD   R2, M[R16]
0000010_00001_00001_00010_0000000000   //   add:   ADD  R1, R1, R2
0100101_00010_00010__0000000000000001  //          SBI  R2, R2, #1
1001000_00000_00010__1111111111111100  //          BNZ  R2, add (-4)
0100010_10010_00000__0000010000000010  //   done:  ADI  R18, R0, #1026
0100000_00000_10010_00001_0000000000   //          ST   M[R18], R1
```

- 完全依賴於指令集 (spec.) 的設定，所以現在新的 CPU 品牌很難普及。PC 至今還在使用 CISC 的原因也大約如此。
- ARM CPU 架構是 RISC。
- 據說 intel 的 CPU 核心架構也改成 RISC 了，只是外層包上了 RISC–CISC 介面。

Demo

- 如果時間允許且沒出意外的話可以 Demo...

Demo

- 如果時間允許且沒出意外的話可以 Demo...
- 潛規則：*ncverilog* 是標準中的標準。

Outline

- Compilers do a lot of works for us.

- Compilers do a lot of works for us.
- OS and drivers helps manage hardwares so much.

Linux 架構示意圖

- Compilers do a lot of works for us.
- OS and drivers helps manage hardwares so much.

Linux 架構示意圖

- High–Level programming is ALL based on Low–Level implementation.