

1. 減少 Dynamic Power, ~~DVFS~~ static power.

DVFS 是利用多個 V_{DD} 與 f

DVFS 是利用多個 V_{DD} 與 f, 利如有些電路需要的供電比較小, 就可以用較小的供電減少 static power

優點: 可以減少 static power

缺點: 做起來複雜, 有 overhead, level shifter. 這額外的面積、cost

clock gating 是將有些暫時不需要的電路的 clock 開起來, 減少 ~~Dynamic Power~~

優點: 減少 dynamic power, 用 Verilog code 寫

缺點: 額外的面積, overhead 滿大的

2. 虽然现在很多很好的 verification techniques, 但很難 reuse
在其他的 stage 和其他 project, 分開的 functional verification process
限制了速度與效率。

因為 FPGA 是可重複編程的, 它可以先做為 prototype 驗証
function 然後不變, 之後再下線就比較不會錯誤, 可以省錢省
時間。

Couple software 執行的環境和 hardware simulator, 在更高 level
的 system 測試, software 通常用 Instruction set Simulator 執行
用 Bus Interface Model 把 software operation 轉成 detailed
pin operation, 之後再丟到 co-simulation 的 kernel。
Hardware 的部分也是跑 simulation 再丟到 co-simulation 的 kernel, 最
後一起分析。

3.

用 code coverage 是希望每一行 code 都會被跑到，以比來驗證 design 到底對不對。

code coverage 會在 Behavioral, RTL, Gate process 的時候用， RTL 也會在 module design, sub-system Integration, System Integration 時候用，可以在上面這些階段先確保 designer 寫的有蟲的部份都有被驗到，這樣之後的 process 比較不會有錯。

4.

Simulation-based

優點: easy to use, low complexity

缺點: 把所有 simulation 漸進會花費龐大的時間, 並且因為現在 design 越來越複雜, design 之間的 interaction 也越來越複雜。

simulation 很難測試, CPU intensive, can handle large system

formal-based

優點: ~~保證~~ 當所有可能的 input 保證 consistency with specification

缺點: memory intensive,

減少 memory 使用量, 加大 system 的 memory, 這樣計算量下降

consolidate BDD, learning based approach in a single environment

5. f1S

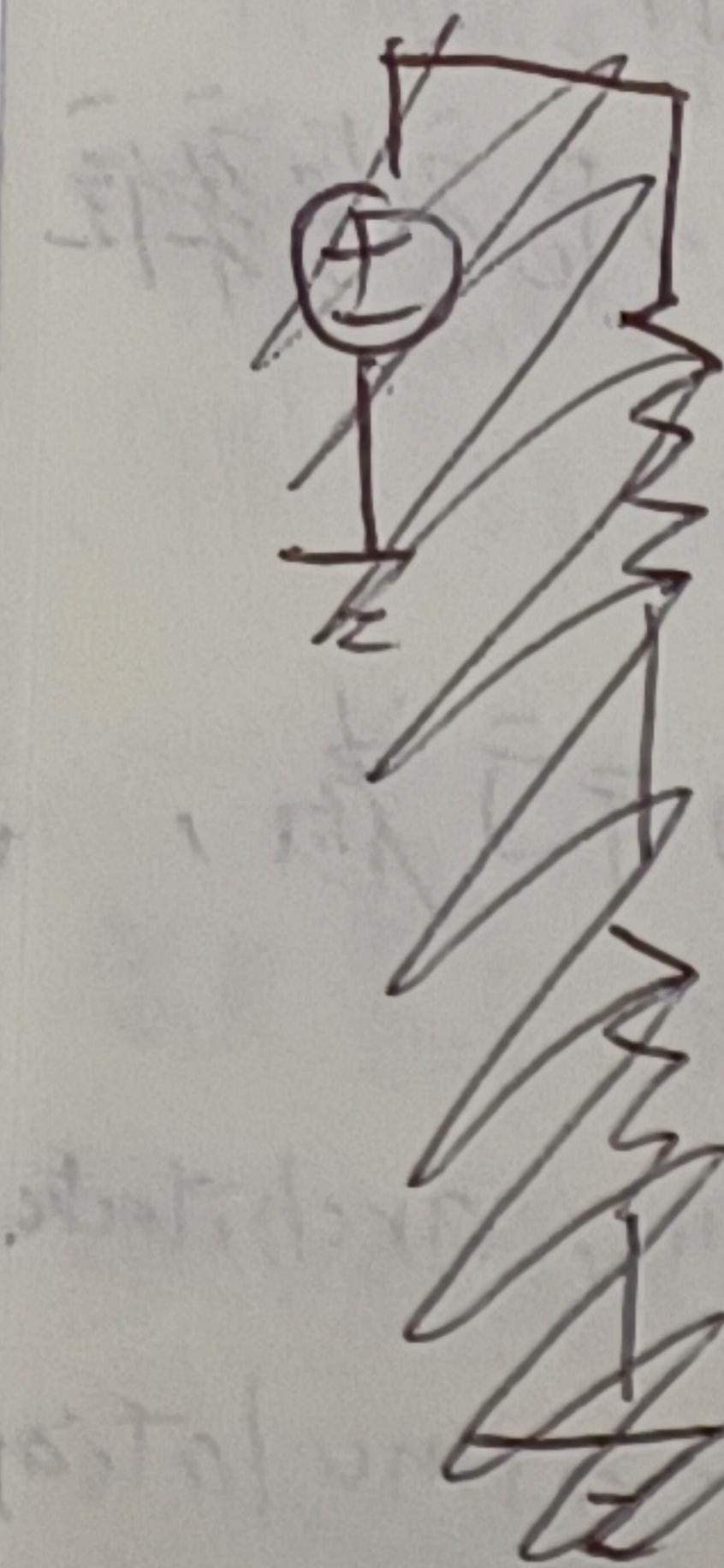
121

Digital 和 analog 分開測的話, 可能會 miss 掉 interaction effect
to coupling, 如果把它們放到 transistor level 測會很久又很複雜

~~有~~ 有 analog behavioral model, 他可以表示 analog circuit 的行為,
可以在更上層的 level 進行 simulation, 例如 system simulation, architecture
verification, 也比較快, 而且可以跟 digital module 一起 simulation
verilog 是 for digital model, 只有 1 跟 0 表示 signal

Verilog-A 是 for analog circuit, 它裡面可以包含 real 的數字
等式, 並且 ~~可以~~ 可以算 KVL, KCL 等等的式子

b.



~~module comparator~~

analog begin

if ($V_{ina} > V_{inb}$)

~~$V_{out} \leftarrow 1$~~

else

~~$V_{out} \leftarrow 0$~~ ; end module

end

module comp (V_{ina}, V_{inb}, out);
input V_{ina} ;
input V_{inb} ;
output out ;

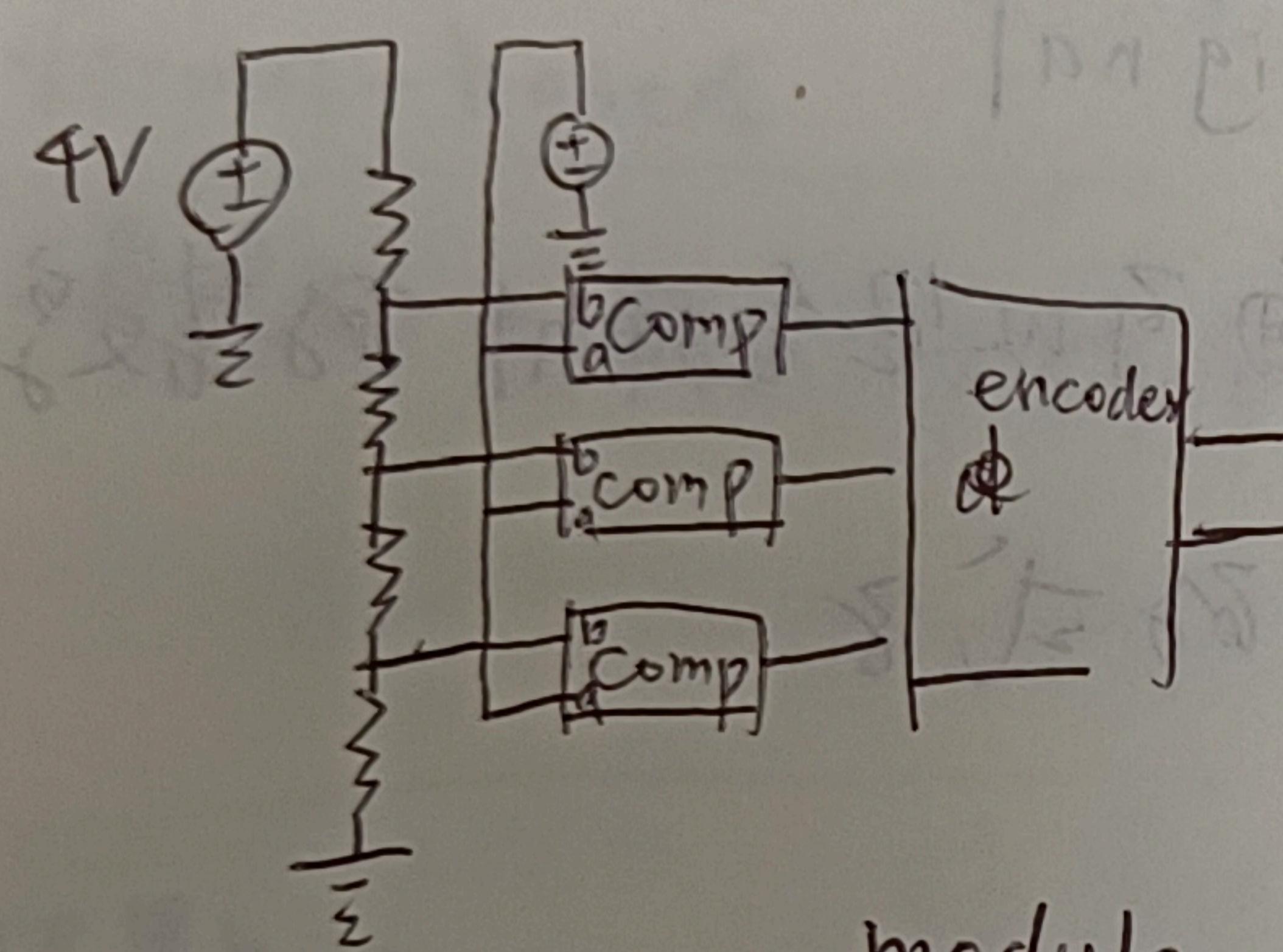
electrical

~~electrical~~

~~real~~

~~out~~

~~out~~



module comp (V_{ina}, V_{inb}, out);

input V_{ina} ;

input V_{inb} ;

output out ;

electrical V_{ina}, V_{inb}, out ;

analog begin

if ($V_{ina} > V_{inb}$)

~~$V_{out} \leftarrow 1.0$~~

else

~~$V_{out} \leftarrow 0$~~ ;

end

end module

7. +15

因為 routing 會有 parasitic effect, 所以結果會不一樣, 通常 post-layout 會比較差。

RC delay, IR drop

用 Bias driven 編程 equation based, 先用 bias driven 找合適的區域算到 V_G, V_S, V_D , g_m, I 等等, 考慮 Capacitance, Body effect Transconductance, 最後 optimization

② 用 worst case parasitic corner, 先 Initial 一版 layout, 然後中 extract layout template, 並在 flow 較早的時候就用 layout-aware synthesis 的時候就把 parasitic effect 利用 layout template 考慮進去, 然後 estimate w, 利用 w 算到 R 跟 C, 用 Worst parasitic corner 算 V_G, V_D, V_{SD} , 有 C/I 可以算 C/I , 然後建表, 之後就可以查表, 之後就丟到 simulation, 最後得到結果

在 layout-aware synthesis

8. +8

把 worst case parasitic corner 考慮進去，~~再用~~ 的時候
就先把它考慮進去，包含 Intra-module 和 Inter-module
先 Initial - 版圖-template，算出 w ，再到 R 跟 C ，~~之後~~ V_D V_S
 V_D, V_S 也可以用 worst case parasitic corner 先算，之後就 zyn 查表
更繼續 simulation 到 result.

果

1.

- (1) ³ reduce dynamic power and static power.
- (2) DVFS dynamically controls the voltage and frequency.
Clock gating turns off the clock when they are not required.
- (3) DVFS - pros : scale the operating voltage or frequency for different tasks to reduce power.
cons : determine which voltage and clock values to support is complicated. increase verification complexity.
Clock gating - pros : turn clocks off when they are not required to reduce power.
cons : level shifter may increase delay, may increase the complexity of timing analysis.

2.

- (1) As complexity increases, it takes more time and more effort to do verification. Also, there are more and more deep submicron effects to consider for modern technology nodes.
- (2) FPGA runs fast and produces actual results. Also, it does not require fabrication, so we can get the designing result quickly on a FPGA. \checkmark
(hardware behavior)
- (3) We can use emulators to verify HW/SW integration for large designs.

3.

(1) ~~Cod~~

(2) ~~Cod~~

~~soft~~

~~use~~

3.

(1) 5

Code coverage uses mathematical methods to examine how much of a code is exercised.

(2) Code coverage could point out areas of code that have not been exercised by the test suites, which helps improve verification reliability. Also, it could help minimize the use of simulation resources.

4.

- (1) Simulation-based - pros: better accuracy, general solution
cons: may take a long time to generate accurate results.
- Formal-based - pros: more efficient and reliable
cons: not suitable for handling large designs due to high complexity, consumes lots of memory.
- (2) simplify the system being verified (using abstraction). utilize simulation results to help improve formal verification efficiency.

5.

- (1) T12 Chip-level simulation takes too much time, there is limited or no use of HDL, designs are not completely systematic ... -2
- (2) With a good analog behavioral model, we could better estimate the outcome of the analog design, leading to better design quality. -1
- (3) Verilog: could only use approximations to model analog behaviors, which limits the model accuracy; however, it provide a choice for faster simulation speed.

Verilog-A: supports non-linear and integro-differential behaviors modeling, which leads to higher accuracy; however, the speedup may be limited due to extra communication between analog and digital simulators.

$$z(t) = \text{cos}(t) \sin(t) + q_{\text{inst}} = q_{\text{inst}}$$

$$z(t) = [t] \sin(t) + q_{\text{inst}} = q_{\text{inst}}$$

$$(0.1, 0.2, 0.3, q_{\text{inst}}) \rightarrow (t, z)$$

6.

module dac_2bit (in1, in2, out);

input in1, in2;

output out;

voltage in1, in2, out;

parameter real threshold = 0.5;

real temp; integer pow2[0:1];

analog begin

@(initial-step) begin

pow2[0] = pow(2,0);

pow2[1] = pow(2,1);

end

temp = 0;

temp = temp + (V(in1) > threshold)? pow2[0] : 0;

temp = temp + (V(in2) > threshold)? pow2[1] : 0;

V(out) <+ transition (temp, 1n, 0.5n, 0.4n);

end.

7. +10

- (1) The synthesis process is not able to take all the parasitics and other non-ideal effects that are presented in the actual layout into account. Thus, there may be some difference after post-layout simulation.
- (2) ① Use more accurate models (equations). If we can consider more non-ideal effects and build a more accurate model, we may have less difference between synthesis and post-layout results. However, more complex equation may limit the runtime.
② Use simulation to get more accurate results. Since simulation is more accurate, we can use equation-based approach to search globally first, and use simulation to gain more accurate results.

8.

+8 We can use a roughly P&R results as feedback, to consider layout-induced parasitic effects more accurately, and modify our current design.

Or we can estimate parasitic effects from the flexible layout template. (P.23 in Lec14)