

CHUYÊN ĐỀ ASIC	ThS. NGUYỄN BÁ HỘI
Đối tượng: sinh viên năm 5, ngành Điện tử	Khoa Điện tử Viễn thông

Sách tham khảo.....	2
Cách thức tính điểm.....	2
Dẫn nhập.....	3
CHƯƠNG 1: Giới thiệu ASIC.....	4
1.1 Các loại ASIC	5
1.1.1. Full-custom ASIC.....	5
1.1.2. Standard-Cell-Based ASIC (CBIC)	6
1.1.3. Gate-array-based ASIC (GA)	8
1.1.4. PLD & FPGA	8
1.1.4.1. PLA & PAL	9
1.2 Qui trình thiết kế ASIC	9
1.3 Kết luận.....	10
CHƯƠNG 2: CMOS logic	11
2.1 CMOS transistor	12
2.1.1. Transistor kênh dẫn loại p.....	15
2.1.2. Bảo hòa vận tốc (velocity saturation)	15
2.1.3. Mức logic.....	15
2.2 Qui trình chế tạo CMOS	16
2.3 Qui luật thiết kế.....	18
2.4 Tế bào logic tổ hợp (Combinational Logic Cell)	20
2.4.1. Định luật de Morgan.....	20
2.4.2. Drive strength	20
2.4.3. TG & MUX	22
2.5 Tế bào logic tuần tự (Sequential Logic Cell)	23
2.5.1. Bộ chốt dữ liệu – latch or D-latch.....	24
2.5.2. Flip-Flop.....	24
2.5.3. Cổng đảo có xung clock - Clocked Inverter	26
2.6 I/O cell	26
2.7 Trình dịch cell - Cell Compiler	26
CHƯƠNG 3: Thiết kế thư viện ASIC	27
3.1 Mô hình trở của transistor	27
3.2 Tự ký sinh	27
3.3 Logical Effort.....	27
3.3.1. Ước tính trễ.....	29
3.3.2. Diện tích logic & hiệu quả logic	30
3.4 Bài tập	31
CHƯƠNG 4: VHDL.....	33

Sách tham khảo

1. Michael J.S. Smith, **Application Specific ICs**, Addison Wesley, 1997
2. Charles H. Roth, **Digital System Design using VHDL**, PWS, 1998
3. Stephen Brown & Zvonko, **Fundamentals of Digital Logic with VHDL Design**, Mc-GrawHill, 2000
4. Neil H.E. Weste & Kamran, **Principles of CMOS VLSI Design – a system prospective**, Addison Wesley, 1993
5. David Johns & Ken Martin, **Analog IC design**, John Wiley & Sons, 1997
6. Kang & Leblebici, **CMOS Digital ICs**, Mc-GrawHill, 1999
7. Allen & Holberg, **CMOS Analog Circuit Design**, Oxford University Press, 2002
8. John P. Uyemura, **Circuit Design for CMOS VLSI**, Kluwer Publisher, 1992
9. Nguyen Quoc Tuan, **Giao trình ngôn ngữ VHDL để thiết kế vi mạch**, 2002

Cách thức tính điểm

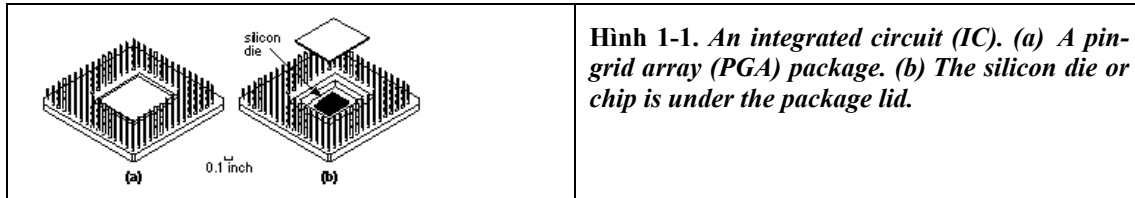
Bài tập:	20%	Thực hành:	20%
Thi cuối kỳ (cho phép dùng tài liệu):	60%		

Dẫn nhập

- Bảng Karnaugh, 2-input NAND, NOR & vẽ mạch CMOS logic tương đương, ký hiệu. Tầm quan trọng của NAND & NOR gates.
- Cổng hỗn hợp (compound gate), n-input gates, AND gate
- **Bài tập 1**
- Tham khảo file [M-chip Disk on chip, filename: NOR_vs_NAND.pdf]: So sánh công nghệ NOR và NAND: kiến trúc của NOR chỉ thích hợp cho các thiết bị lưu trữ từ 1 – 4MB, NOR cho hiệu suất đọc cao nhưng thời gian xóa và thời gian lập trình lớn, nên không thích hợp cho các thiết bị lưu trữ yêu cầu dung lượng và tốc độ cao như hiện nay. NAND có được các tính năng vừa nêu, dung lượng từ 8 – 512 MB cộng với giá cả phải chăng hơn. Bù lại, các nhà chế tạo phải đương đầu với giao diện không chuẩn (non-standard interface) và sự quản lý phức tạp (complicated management) của NAND

CHƯƠNG 1: Giới thiệu ASIC

ASIC (Application Specific IC). IC and PGA package (Pin Grid Array) (hình 1.1).



Hình 1-1. An integrated circuit (IC). (a) A pin-grid array (PGA) package. (b) The silicon die or chip is under the package lid.

Tính toán kích cỡ IC: theo số lượng cổng (logic-gate hay transistor) bên trong IC. Đơn vị tính kích cỡ IC là NAND hoặc NOR gate. Ví dụ: 100k-gate = 100.000 two-input NAND gates. 2-input NAND gate = 4 CMOS transistors. Tương tự cho NOR gate. (xem chương dẫn nhập)

Các giai đoạn phát triển của công nghệ tích hợp: SSI (thập niên 70), MSI, LSI, VLSI, ULSI. (SSI với vài chục transistor tức cỡ 1-10 gates, LSI có thể chế tạo microprocessor, thuật từ VLSI (phổ biến) = ULSI (Nhật))

TTL (ECL – emitter coupled logic)	Đầu thập niên 70	Bipolar IC <ul style="list-style-type: none"> Tốn năng lượng Giá thành cao Kích cỡ lớn
NMOS	Thập niên 70	MOS IC <ul style="list-style-type: none"> Metal gate nMOS, chưa có pMOS Ít các bước masking Mật độ cao hơn (denser) Tiêu tốn ít năng lượng (consumed less power) Thị trường MOS IC
CMOS	Thập niên 80	CMOS IC <ul style="list-style-type: none"> Đột phá: Polysilicon Gate cho phép tích hợp nMOS & pMOS trên cùng IC Tiêu tốn ít năng lượng hơn nữa Polysilicon cho phép đơn giản quá trình chế tạo dẫn đến thu nhỏ kích cỡ IC

Bipolar & BiCMOS ICs vẫn được sử dụng trong các ứng dụng điện thế cao (s/v CMOS) như điện tử công suất, xe hơi, mạch điện thoại ...

Feature size: đặc trưng bởi λ ; $\lambda = \frac{1}{2}$ smallest transistor size; VD: $\lambda = 0,25\mu\text{m}$ tương ứng transistor nhỏ nhất có kích cỡ $0.5 \mu\text{m}$ (liên hệ cấu tạo CMOS transistor và công nghệ chế tạo)

Thông thường, xây dựng hệ thống vi điện tử (microelectronic system) sử dụng các thành phần chuẩn - “standard parts” hay IC chuẩn - “**standard ICs**”. Sau sự ra đời của VLSI những năm 80, ta có thể xây dựng mọi thứ trên một IC đơn cho các ứng dụng chuyên dụng khác nhau (customized to a particular system) → “**custom ICs**”. Tất nhiên là không phải trường hợp nào cũng thích hợp. Nguyên tắc là định nghĩa yêu cầu

bài toán (xác định design entry), sau đó xây dựng một số phần sử dụng standard IC, phần còn lại sử dụng custom IC → giá rẻ, tăng độ tin cậy.

Custom IC là hoàn toàn không cần thiết đối với bộ nhớ chẳng hạn.

IEEE Custom IC Conference (CICC) → custom IC được phát triển mạnh mẽ cho vô số các ứng dụng khác nhau → thuật ngữ ASIC, IEEE International ASIC Conference cho riêng ASIC.

<i>Not ASIC</i>	<i>ASIC</i>	<i>Lưu ý tính</i>
ROM DRAM, SRAM Microprocessor TTL, TTL-equivalent IC ở các mức tích hợp SSI, MSI, LSI → qui tắc: có thể tìm thấy trong “data book”	Gấu đồ chơi nói được Satellite chip Chip đảm nhận việc giao tiếp giữa workstation CPU với bộ nhớ Chip chứa microprocessor cùng với thành phần logic khác → chuyên dụng (Application Specific IC)	PC chip Modem chip → sản xuất rộng rãi (ASSPs)

Nhận dạng người:

- Gương mặt
- Các đặc điểm vật lý
- ...

Nhận dạng ASIC:

- Các đặc điểm vật lý
- Giá thành
- Phương pháp thiết kế ASIC
- ...

→ các loại ASIC

1.1 Các loại ASIC

Nêu các khái niệm Wafer, Mask layer, Interconnect.

Full-custom ASICs <ul style="list-style-type: none"> ▪ logic cells & mask layers được thiết kế theo yêu cầu user ▪ giá thành cao ▪ 8 tuần chế tạo (không kể thời gian thiết kế)
Semi-custom ASICs <ul style="list-style-type: none"> ▪ logic cells được thiết kế sẵn → cell library ▪ một vài hoặc tất cả mask layers được thiết kế theo yêu cầu user ▪ standard-cell-based ASICs ▪ gate-array-based ASICs
Programmable ASICs <ul style="list-style-type: none"> ▪ logic cells & mask layers đều được thiết kế sẵn ▪ PLDs ▪ FPGAs (what we can do in Danang!!!)

1.1.1. Full-custom ASIC

Đặc điểm:

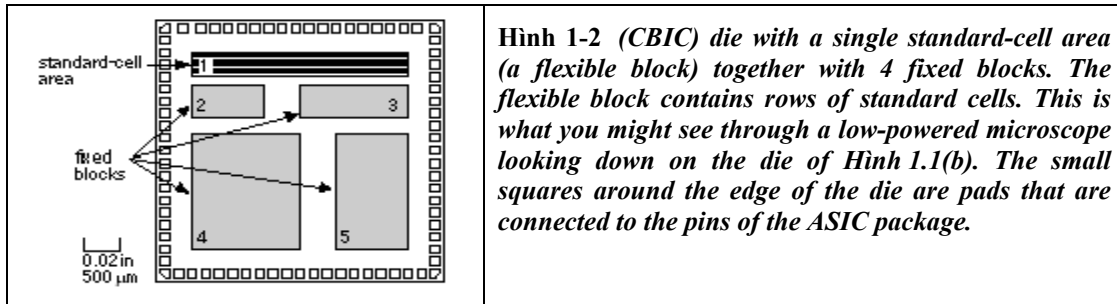
- logic cells & mask layers được thiết kế theo yêu cầu user
- giá thành cao
- thời gian chế tạo 8 tuần (không bao gồm thời gian thiết kế)

1.1.2. Standard-Cell-Based ASIC (CBIC)

Nêu các khái niệm:

Standard cell = logic cell = cell (AND, OR, MUX, Flip-Flop, Latch).

Megacell = full-custom block = System Level Macro (SLM) = fixed block = core = Functional Standard Block (FSB). VD: SRAM, SCSI Controller, MPEG Decoder...



Đặc điểm CBIC:

- mask layers được thiết kế theo yêu cầu user
- vì vậy cell & megacell có thể đặt bất kỳ đâu và trên cùng 1 chip

Ưu:

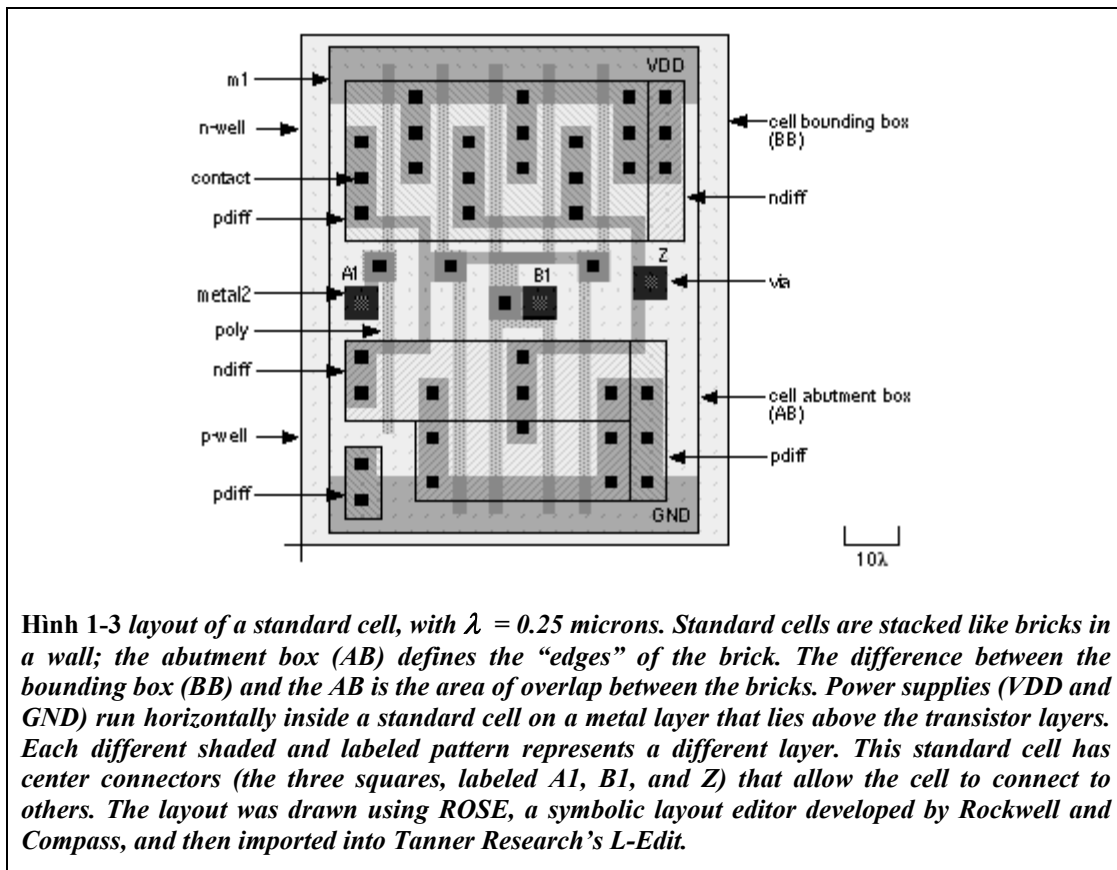
- Cell được thiết kế sẵn (predesigned) → mỗi cell được thiết kế tối ưu độc lập
- Cell được kiểm tra (pretested) → giảm rủi ro
- Cell được đặc tả rõ (precharacterized) → giảm giá thành
- tiết kiệm thời gian thiết kế

Nhược:

- Thời gian thiết kế hay chi phí mua thư viện cell
- Thời gian chế tạo các mask layer
- Thời gian chế tạo: 8 tuần (không bao gồm thời gian thiết kế)

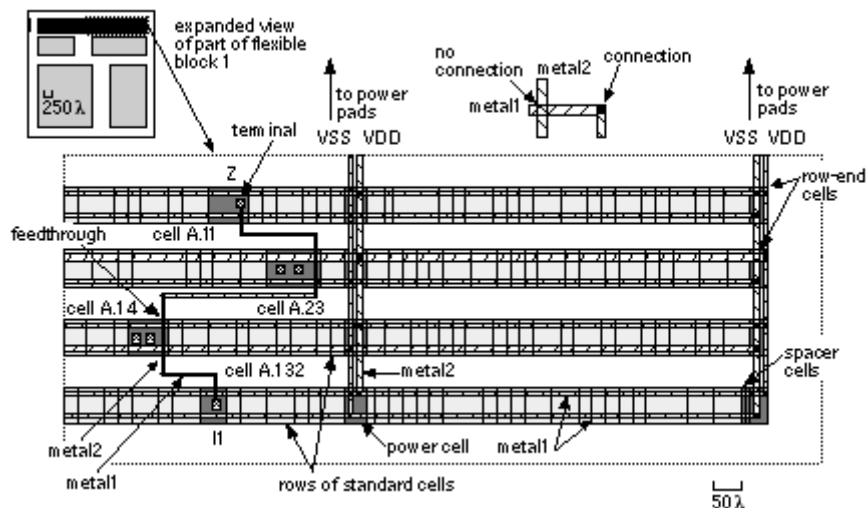
Cell-based ASIC (CBIC)	Gate-based ASIC (GA)
Điểm chung: Predesigned cells	
Có thể thay đổi kích cỡ transistor trong cell để tối ưu hóa tốc độ và hiệu suất	Kích cỡ transistor cố định (fixed cell)
→ Sự thỏa hiệp giữa diện tích (area) và hiệu suất (performance) ở tầng thư viện	→ Sự thỏa hiệp giữa diện tích (area) và hiệu suất (performance) ở tầng silicon

ASIC tiên tiến dùng 2 đến 3 lớp kim loại (metal layer) hoặc nhiều hơn cho interconnect. Metal 1: power bus. Metal 2: input hay output cells. Xem hình 1.3.



Các khái niệm:

Feedthrough:	đường dẫn kim loại xuyên qua cell
Spacer cell:	hiệu chỉnh chiều dọc các hàng cell
Row-end-cell:	kết nối nguồn cho các hàng khác nhau
Power-cell:	dùng khi cell-row quá dài



Hình 1-4 Routing the CBIC (cell-based IC) shown in hình 1.2. The use of regularly shaped standard cells, such as the one in hình 1.3, from a library allows ASICs like this to be designed automatically. This ASIC uses two separate layers of metal interconnect (metal1 and metal2) running at right angles to each other (like traces on a printed-circuit board). Interconnections between logic cells uses spaces (called channels) between the rows of cells. ASICs may have three (or more) layers of metal allowing the cell rows to touch with the interconnect running over the top of the cells.

Datapath:

Khi nhiều tín hiệu đi qua một bus dữ liệu thì các logic cell không còn hiệu quả, khi đó, datapath được sử dụng. Tạo ra datapath bằng **datapath compiler** từ các nhà SX. **Datapath library** bao gồm các *datapath cell* như là: bộ cộng - adder, bộ trừ - subtracter, bộ nhân - multiplier & khối logic số học đơn giản – simple ALU. **Ưu:** kết nối các *datapath cell* để tạo nên datapath **thông thường** cho ra layout chặt hơn (tốn ít diện tích) & hoạt động nhanh hơn (so với standard-cell hay gate-array).

1.1.3. Gate-array-based ASIC (GA)

SV tự đọc sách.

1.1.4. PLD & FPGA

- Logic cell và mask layer có sẵn (không theo yêu cầu user)
- Interconnect khả trình
- Ma trận các macrocell bao gồm các PAL + FF hoặc Latch
- Thời gian thiết kế hoàn chỉnh khá nhanh (vài giờ)

Ví dụ:

- Field-programmable: PROM, EPROM, EEPROM, UVPROM
- Mask-programmable: Mask-programmable ROM (Masked ROM)

Field-programmable: các kết nối dùng chuyển mạch lập trình được (cấu chi chẳng hạn, CMOS transistor) & vì vậy chậm hơn các kết nối cứng nhưng có ưu điểm là rẻ khi SX với số lượng nhỏ và thời gian lập trình tức thì.

Mask-programmable: các kết nối bên trong được thực hiện bằng phân cứng khi SX có nhược điểm là lập trình mất vài tháng, song bù lại giá thành giảm nếu SX với số lượng lớn.

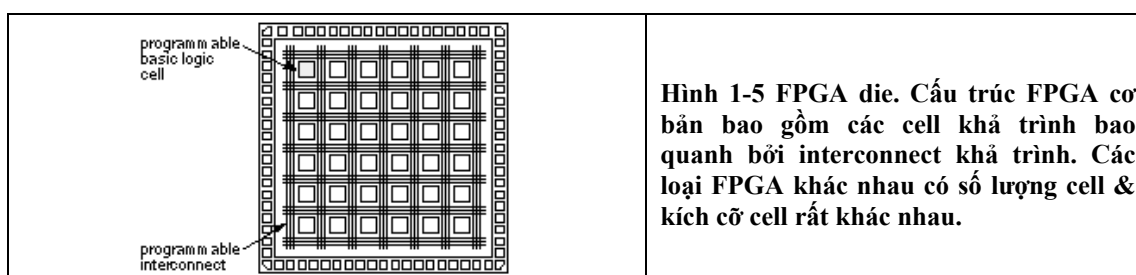
PLD: gồm khối cổng AND nối với khối cổng OR. Mạch logic thực hiện trong PLD theo dạng tổng của tích (sum-of-product).

Các loại PLD:

- PLD cơ bản: PAL (Khối AND khả trình, khối OR cố định)
- PLD linh hoạt: PLA (Khối AND và OR đều khả trình). PLA có thể là mask-programmable hay field-programmable.

Cả hai loại PLD trên cho phép thực hiện các mạch logic tốc độ cao. Tuy nhiên cấu trúc đơn giản của nó chỉ cho phép hiện thực các mạch logic nhỏ.

Các PLD phức tạp (complex PLD - CPLD) được biết đến như những FPGA.



1.1.4.1. PLA & PAL

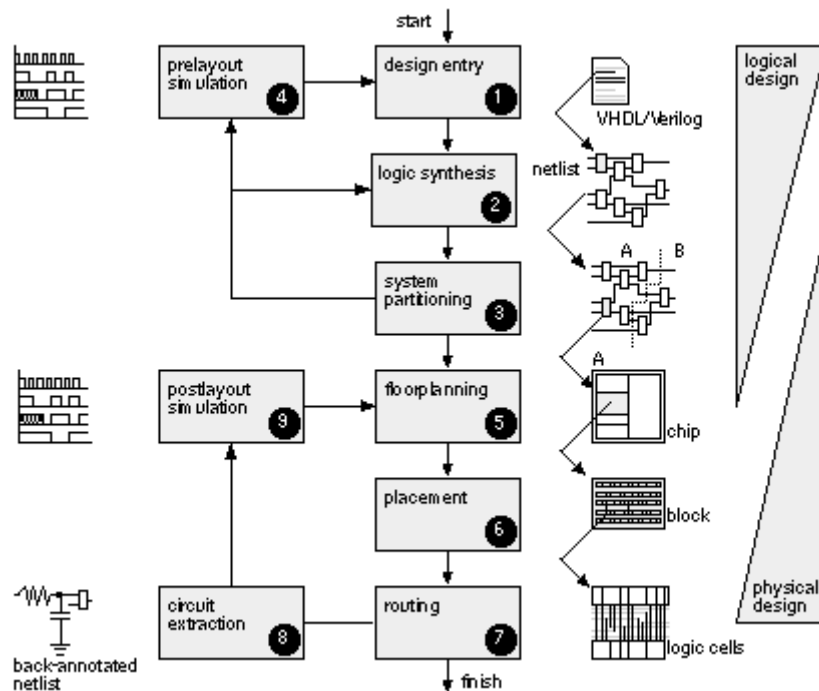
Cấu trúc PLA: Mạng logic khả trình.

Tìm bảng PLA hàng tối thiểu

Cấu trúc PAL: Logic mạng khả trình, là tr.h riêng của PLA - mạng OR cố định.

Bài tập 2

1.2 Quy trình thiết kế ASIC



Hình 1-6 ASIC design flow

Better impress this flow on the memory by explaining in comparison with building construction.

1. **Mô tả bài toán:** sử dụng ngôn ngữ mô tả phần cứng HDL (VHDL hay Verilog)
(VHDL by Department of Defense in 1980s and standardized by IEEE in 1993 - Verilog is created by Cadence in 1989 and standardized by IEEE in 1995)
2. **Tổng hợp logic:** dùng HDL và công cụ tổng hợp logic để xây dựng netlist – là sự mô tả các tế bào (cell), các khối (block) và kết nối (interconnect) giữa chúng
3. **Phân chia hệ thống:** chia hệ thống lớn thành các phần thích hợp
4. **Mô phỏng tiền layout:** kiểm tra tính đúng đắn của thiết kế (tiền layout = sơ đồ mạch logic – chỉ gần đúng với thực tế)
5. **Sắp xếp các khối trên chip:** sắp xếp các khối của netlist trên chip. Nên xem xét cả khía cạnh vật lý và logic khi thiết kế bước này
6. **Bố trí cell:** định vị cell bên trong khối
7. **Thiết kế tuyến:** kết nối giữa các cell và các khối
8. **Kiểm tra tính hợp lý của bước 7:** tính toán trở kháng và dung kháng lớp interconnect

9. **Mô phỏng hậu layout:** kiểm tra khả năng làm việc ổn định của toàn bộ thiết kế trong trường hợp có thêm tải từ lớp interconnect (hậu layout = sơ đồ mạch thực tế)

Các bước thiết kế 1 - 5: logic.

Các bước thiết kế 5 – 9 : vật lý.

1.3 Kết luận

ASIC: thay vì phải xem xét nhiều khía cạnh trong khi thiết kế chế tạo IC chuẩn thì công nghệ ASIC cho phép tối ưu hóa thiết kế theo một mục đích cụ thể (specific task) nên sẽ cho hiệu suất cao hơn, cụ thể là cho phép lưu giữ **lượng mạch logic** (*chú ý chỉ là mạch logic*) lớn hơn so với các chip chuẩn cùng kích thước. Bên cạnh tính tin cậy cao thì IC đơn lẻ còn chiếm dụng ít không gian hơn trên bo mạch in, kéo theo giá thành rẻ hơn so với 1 hệ thống có cùng mục đích sử dụng nhiều IC chuẩn. Trình tự thiết kế ASIC theo các bước : 1. Thiết kế logic (logic design) 2. Chọn kỹ thuật thích hợp thiết kế mạch vật lý (physical design) 3. Chế tạo chip (fabrication) bởi công ty chuyên nghiệp.

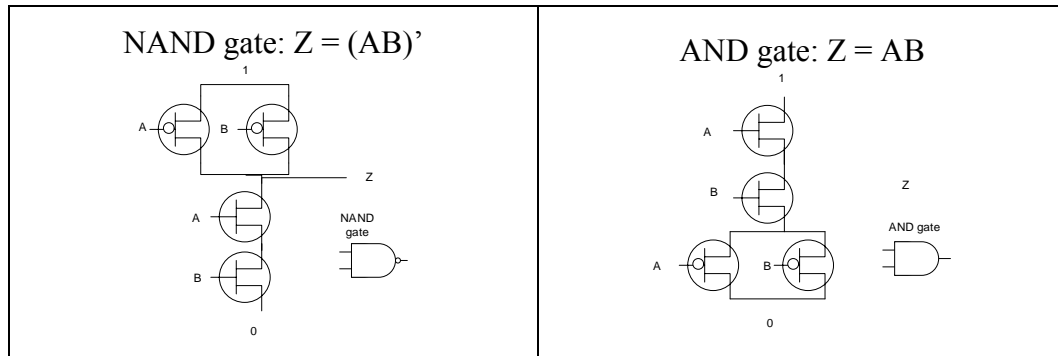
ASIC khả trình: CPLD hay FPGA: chứa các chuyển mạch lập trình được nhiều lần (các chuyển mạch sử dụng cho cả cell khả trình và **interconnect** khả trình). Các PLA thường được xem là thành phần cơ bản của FPGA.

*CPU Pentium 4 chứa 55 triệu cổng chế tạo bằng công nghệ 80-130nm. Với các vật liệu mới, chip ngày càng được thu nhỏ thì khi đó lượng điện thoát ra khỏi bóng bán dẫn càng lớn, do đó tạo ra sức nóng lớn hơn và khiến các transistor dễ bị hỏng (tựa như phiên tỏa nhiệt càng nhỏ thì khả năng tản nhiệt càng thấp). Theo ghi nhận của các nhà khoa học ĐH Maryland (Mỹ) thì silicon đã có hậu duệ: đó là **carbon nanotube**. Chất này có độ dẫn điện mạnh gấp 70 lần silicon, đồng thời cũng cho cường độ dòng điện lớn hơn. Trong khi theo hãng Toyota và Denso thì chất mới là **Silicon Carbua (SiC)**.*

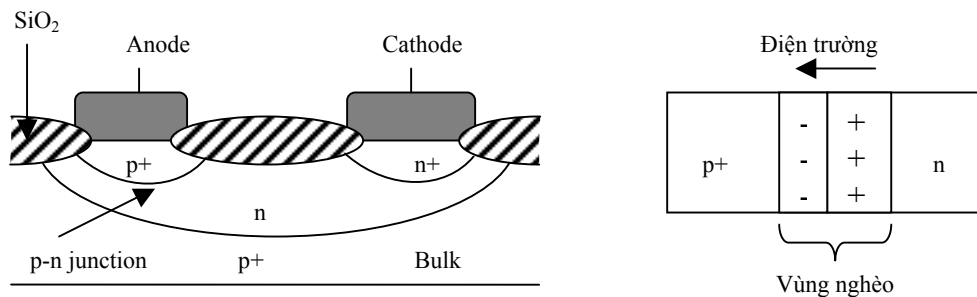
Cuối năm 2003, Intel thông báo sẽ cho ra đời chip 45 - 65nm trong thời gian tới, thì vào tháng 09 năm 2004, chip mới với công nghệ 65nm (1 tỷ transistor) đã ra đời. Intel không cho biết tên chính xác loại vật liệu mới.

CHƯƠNG 2: CMOS logic

Dẫn nhập: chúng ta đã biết tới việc sử dụng NAND hoặc NOR như các thành phần cơ bản trong chế tạo IC. **Vậy tại sao lại phải là NAND hoặc NOR?** Cấu thành NAND hay NOR theo công nghệ CMOS cơ bản tiêu tốn 4 transistor (2 nMOS và 2 pMOS) trong khi các cổng AND, OR cũng có thể chế tạo từ 4 transistor!!!



Diode và vùng nghèo: các tiếp xúc kim loại thường nối với vùng p+ và n+ (có mức kích tạp cao hơn so với p và n) để tránh các diode schottky. Vùng p+ có lượng lớn các ion dương tự do còn trong vùng n+ có lượng lớn các ion âm tự do. Lỗ trống trong p+ có khuynh hướng khuếch tán sang phía n trong khi electron trong n lại khuếch tán sang p+ (giống như các loại gas trộn lẫn vào nhau). Sự khuếch tán vì vậy làm giảm sự tập trung của các ion tự do trong vùng tiếp xúc. Khuếch tán của electron từ n sẽ làm x/h vùng điện tích + bên phía n tại vùng t/x, ngược lại, khuếch tán của lỗ trống từ p+ sẽ làm x/h vùng điện tích - bên phía p tại vùng t/x. Sự khuếch tán của các ion tự do này hình thành nên vùng t/x nghèo (depletion region).



Mặt cắt diode p-n

Mô hình đơn giản – độ rộng vùng nghèo lớn hơn cho phía kích tạp thấp n

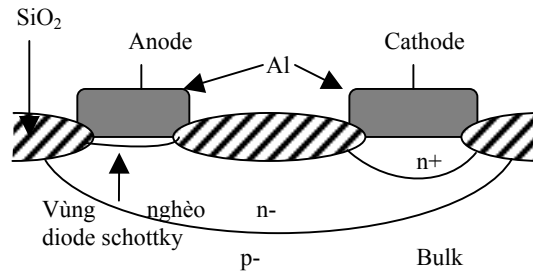
Hình 2-1 Diode p-n

Diode schottky: khi có tiếp xúc kim loại trực tiếp tại bề mặt chất bán dẫn kích tạp nhẹ (n, n- hay p, p-), điều này khiến cho đặc tính diode có thay đổi so với p-n junction diode thông thường,

1. V_D nhỏ hơn bình thường, chỉ 0.3-0.5V s/v 0.6-0.8V đ/v silicon p-n diode, là do sự khác biệt về tính năng làm việc giữa kim loại và n- là cao hơn s/v giữa kim loại và n+.

2. Dòng trong diode schottky chỉ do hạt dẫn đa số tạo nên (electrons). Khi diode phân cực thuận sẽ không còn sự tích điện của hạt dẫn thiểu số trong vùng n-, hay nói $C_d = 0$ (depletion cap) trong mô hình tương đương tín hiệu nhỏ. Điều này làm cho diode đáp ứng nhanh hơn, đặc biệt là khi turn-off - vì không cần phải xả điện tích thiểu số.

GaAs được dùng chế tạo diode schottky.

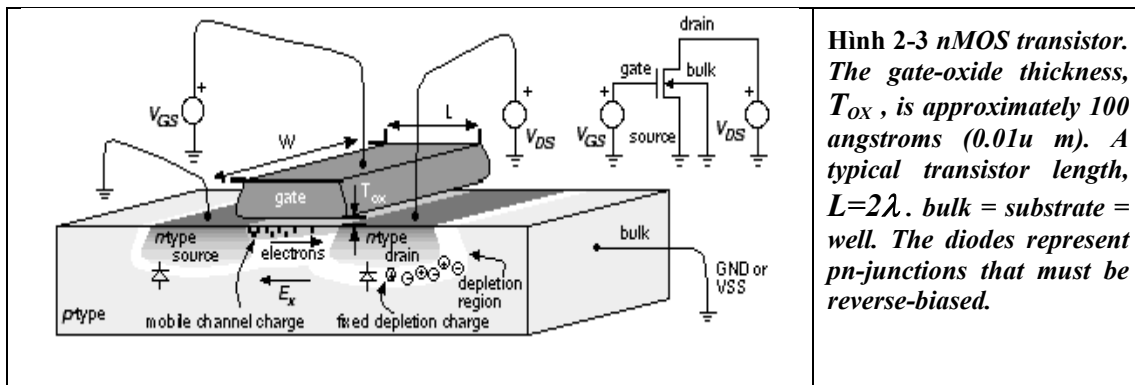


Mặt cắt diode Schottky

Hình 2-2 Diode Schottky

Vai trò n+ cho các tiếp xúc giữa bán dẫn kích tạp nhẹ với kim loại?

2.1 CMOS transistor



Hình 2-3 nMOS transistor. The gate-oxide thickness, T_{ox} , is approximately 100 angstroms (0.01 μm). A typical transistor length, $L=2\lambda$. bulk = substrate = well. The diodes represent pn-junctions that must be reverse-biased.

Khi không có các tác động của điện thế bên ngoài, miền không gian giữa cực D và S không dẫn điện (không có sự di chuyển của các điện tích - electron). Để kích dẫn transistor MOS loại kênh n, chúng ta cần đưa vào cực G điện thế V_{GS} dương lớn hơn điện thế ngưỡng V_{tn} - threshold voltage cỡ 0.5V. Điện thế này làm hình thành 1 kênh dẫn rất mỏng (50\AA , $1\text{\AA} = 10^{-10}m$) bên dưới bề mặt cực cửa G. (MOS transistor tạo ra dòng rỉ vài micro ampe khi V_{GS} bé hơn V_{tn} , tạm thời không xem xét tới TH này). **Transistor MOS có thể dẫn mà không có dòng chạy qua. Dòng chỉ có khi đặt điện thế V_{DS} hợp lý vào 2 cực D và S. V_{DS} là dương với nMOS.**

Well (bulk, substrate or tub): kết nối với nơi có điện thế nhỏ nhất, ký hiệu GND hay VSS, nhằm **đảm bảo phân cực ngược** cho các diode hình thành bởi các tiếp giáp p-n của *bulk - drain* hay *bulk - source*. **Mũi tên** ở terminal 4 - **bulk** biểu diễn chiều của các diode này.

Dòng qua transistor (A) = điện tích (C) / thời gian (s)

Nếu gọi Q là tổng điện tích trong kênh dẫn, t_f là thời gian các điện tử di chuyển từ S sang D (nơi có điện thế thấp sang nơi có điện thế cao), thì dòng I_{DSn} có giá trị:

$$I_{DSn} = \frac{Q}{t_f} \quad (0.1)$$

Tìm t_f : Thật vậy, theo Ohm's Law:

$$v = -\mu_n E \quad (0.2)$$

μ_n độ di động điện tử = $500-1000 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$. E (Vm^{-1}) trường điện từ gây bởi V_{DS} . Để đơn giản, với E ta chỉ xét thành phần ngang E_x , bỏ qua thành phần dọc E_y . $E_x = -V_{DS} / L$ tính từ D tới S. L: chiều dài gate, $= 2\lambda$. Điện tử di chuyển qua đoạn đường L với vận tốc $v = -\mu_n E$ tốn khoảng thời gian:

$$t_f = \frac{L}{v_x} = \frac{L^2}{\mu_n V_{DS}} \quad (0.3)$$

Tìm Q: kênh dẫn và gate tạo nên 2 bề mặt của 1 tụ điện mà chất cách điện là dioxide. Với tụ tuyến tính $Q=CV$. Với TH chúng ta, kênh dẫn là bề mặt dẫn điện phi tuyến tính - điện tích chỉ xuất hiện trên kênh khi V_{GC} lớn hơn V_m . Với tụ phi tuyến thể này, ta có:

$$Q = C(V_{GC} - V_m) \quad (0.4)$$

Trong đó $C = \frac{WL\epsilon_{ox}}{T_{ox}} = WLC_{ox}$

ϵ_{ox} là hằng số điện môi gate-oxide. C_{ox} là điện dung đơn vị.

Ta có $V_{GC} = V_{GS}$ tại S và $V_{GC} = V_{GS} - V_{DS}$. Nếu giả sử V_{GC} là hàm tuyến tính theo x (0 \rightarrow L), thì giá trị trung bình của điện tích:

$$Q = C \left[(V_{GS} - V_m) - \frac{1}{2} V_{DS} \right] \quad (0.5)$$

biểu diễn Q theo các tham số transistor:

$$Q = WLC_{ox} \left[(V_{GS} - V_m) - \frac{1}{2} V_{DS} \right] \quad (0.6)$$

Cuối cùng ta có công thức cho I_{DS} :

$$\begin{aligned} I_{DS} &= \frac{Q}{t_f} = \frac{W}{L} \mu_n C_{ox} \left[(V_{GS} - V_m) - \frac{1}{2} V_{DS} \right] V_{DS} \\ &= \frac{W}{L} k_n' \left[(V_{GS} - V_m) - \frac{1}{2} V_{DS} \right] V_{DS} \\ &= \beta_n \left[(V_{GS} - V_m) - \frac{1}{2} V_{DS} \right] V_{DS} \end{aligned} \quad (0.7)$$

trong đó, **tham số hỗ dẫn** (transconductance parameter) k_n' :

$$k_n' = \mu_n C_{ox} \quad (0.8)$$

Hệ số độ lợi:

$$\beta_n = k_n' \frac{W}{L} \quad (0.9)$$

với $\frac{W}{L}$ là **hệ số hình dáng** (shape factor).

Vùng tuyến tính $V_{GS} > V_{tn}, V_{DS} \leq V_{GS} - V_{tn} = V_{DS(sat)}$: *linear region - triode region*

(0.7) biểu diễn hoạt động transistor trong *vùng tuyến tính*.

Vùng bão hòa $V_{GS} > V_{tn}, V_{DS} > V_{GS} - V_{tn} = V_{DS(sat)}$: *saturation region – active region*

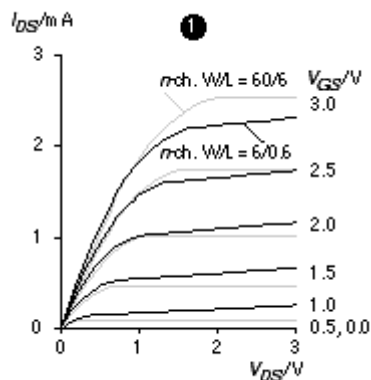
khi V_{DS} vượt quá giá trị $V_{DS(sat)}$ thì giá trị V_{GC} không đủ cho việc duy trì kênh dẫn, $V_{GC(sat)} = V_{GS} - V_{DS(sat)} \rightarrow V_{GC} \leq V_{tn}$ khi $V_{DS} \geq V_{DS(sat)}$, hay nói kênh dẫn bị thắt tại cực D.

Tính xấp xỉ:

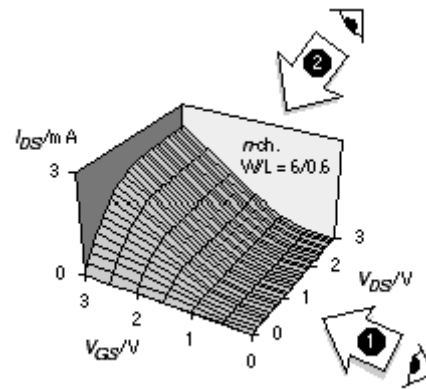
$$I_{DS-sat} = \frac{W}{2L} k'_n (V_{GS} - V_{tn})^2 = \frac{\beta_n}{2} (V_{GS} - V_{tn})^2 \quad (0.10)$$

$\neq V_{DS}$

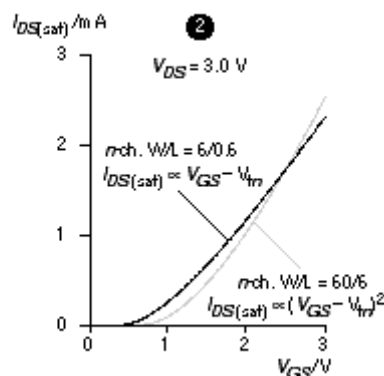
Hình vẽ dưới đây chỉ ra đặc tuyến $I_{DS}-V_{DS}$ cho công nghệ CMOS 0.5 μm (gọi là **G5**).



(a) *kênh dẫn ngắn*, $W = 6 \mu m$ & $L = 0.6 \mu m$ (*dậm*)
kênh dẫn dài ($W = 60 \mu m$, $L = 6 \mu m$)



(b) *đặc tuyến 6/0.6 dạng bề mặt*



(c) *quan hệ I_{DS} và V_{GS} đ/v kênh dẫn dài theo luật hàm mũ trong vùng bão hòa ($V_{DS} = 3 V$). kênh dẫn ngắn cho kết quả tuyến tính hơn do hiện tượng bão hòa vận tốc.*

Thông thường, tất cả transistor trong ASIC là loại kênh dẫn ngắn

Hình 2-4 Đặc tuyến cho CMOS kênh dẫn n công nghệ 0.5 μm (G5)

Vẽ layout cổng CMOS NAND2, chỉ ra thứ tự thiết kế các layer?

2.1.1. Transistor kênh dẫn loại p

Có 2 cách biểu diễn, hoặc theo trị tuyệt đối, hoặc theo dấu âm như sau

$$I_{DS} = -\frac{W}{L} k'_p \left[(V_{GS} - V_{tp}) - \frac{1}{2} V_{DS} \right] V_{DS} \quad V_{DS} > V_{GS} - V_{tp} \quad (0.11)$$

$$I_{DS} = -\frac{\beta_p}{2} (V_{GS} - V_{tp})^2 \quad V_{DS} < V_{GS} - V_{tp}$$

trong đó $V_{tp}, V_{DS}, V_{GS} < 0$

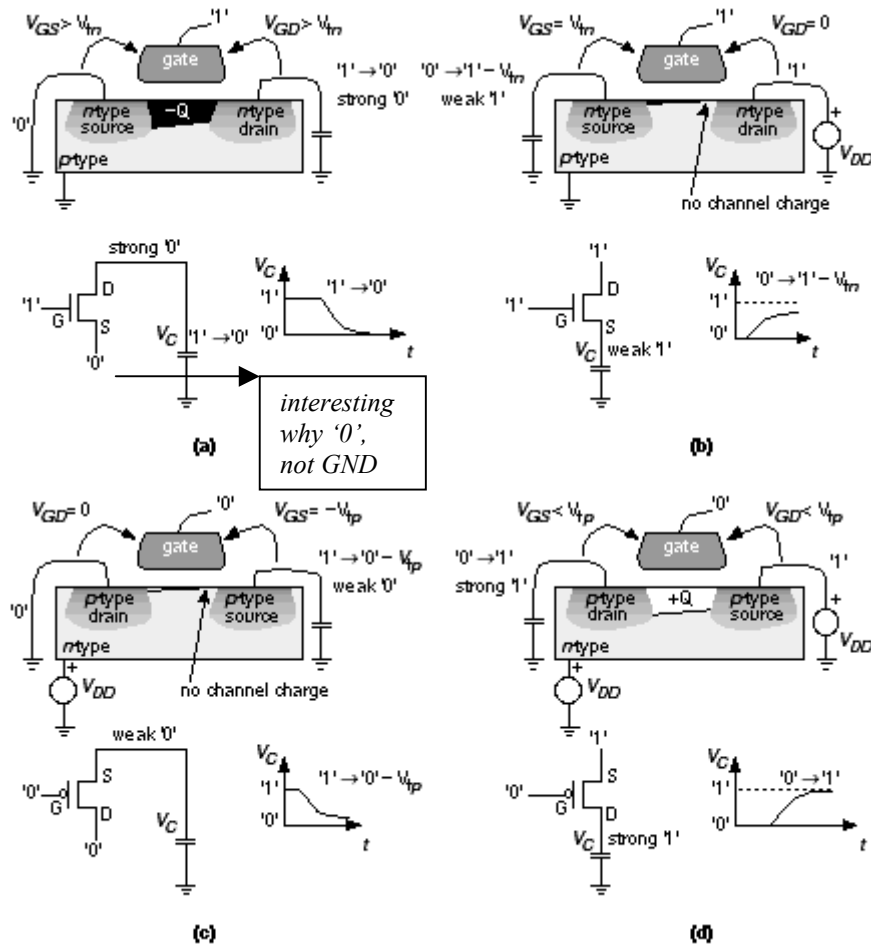
2.1.2. Bảo hòa vận tốc (velocity saturation)

Với transistor kênh dẫn ngắn, giá trị thực của I_{DS-sat} thường gấp 2 lần giá trị trong (0.10). Vì 3 lý do:

1. $V_m \neq const$,
2. $L_{eff} < L_{real}$,
3. BT $v = -\mu_n E$ không còn đúng với điện trường cao, $v_{maxn} = 10^5 Vm^{-1}$ khi điện trường vào cỡ $10^6 Vm^{-1}$, tức electron bão hòa vận tốc (velocity saturated)

Khi này $t_f = \frac{L_{eff}}{v_{maxn}}$ và $I_{DS} = \frac{1}{2} W v_{maxn} C_{ox} (V_{GS} - V_{tn}) \notin L$; $V_{DS} > V_{DSsat}$ (0.12)

2.1.3. Mức logic



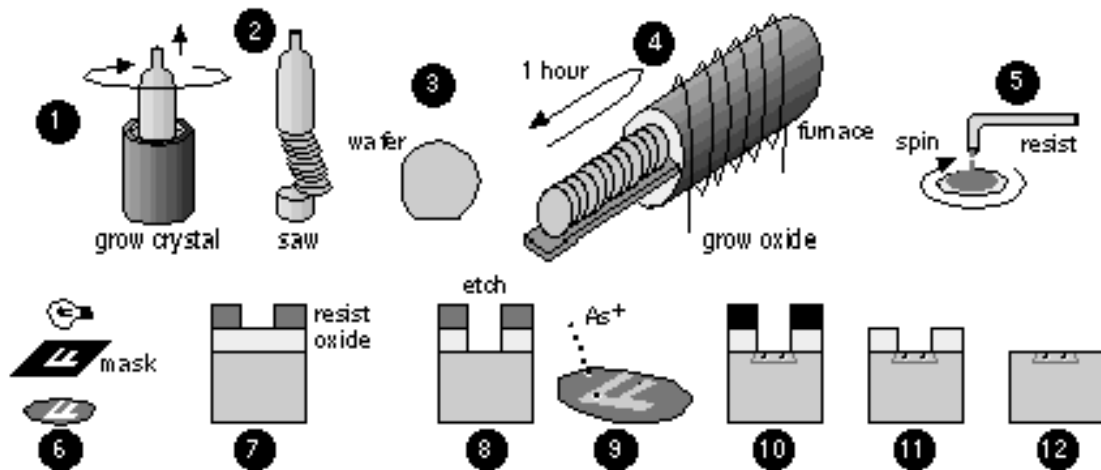
Hình 2-5 Các mức logic khỏe và yếu (a) ‘0’ khỏe (b) ‘1’ yếu (c) ‘0’ yếu (d) ‘1’ khỏe

Chú ý việc vẽ hình, thoát đầu chỉ nhận biết cực G, còn cực D và S chưa xác định. Hình a. logic 1 tại G, logic 0 tại S (nhận biết cực này là S vì điện thế thấp nhất), khiến transistor dẫn dòng (electron di chuyển từ S tới D). Nếu thoát đầu D là logic 1 thì nMOS xả điện tích của tụ nối vào D (tụ hình thành bởi cell khác). Khi tụ xả hoàn toàn, $V_{GS} = V_{GD} = V_{DD}$. Transistor dẫn rất mạnh nhưng không có dòng chảy qua ($V_{DS} = 0 \rightarrow v \rightarrow t_f \rightarrow I_{DS}$). Ngõ ra D lúc này chuyển sang logic 0 – và đây thực sự là logic 0 đúng nghĩa – **logic 0 khỏe**. Giải thích tương tự, hình b cho logic 1 yếu, hình c cho logic 0 yếu, hình d cho **logic 1 khỏe**.

Kết luận: nMOS cung cấp logic 0 khỏe, logic 1 yếu. pMOS thì ngược lại. Ghi nhớ nguyên tắc này khi thiết kế mạch.

Nêu các cell có ứng dụng nguyên tắc trên, phân tích hoạt động của cell để làm rõ ưu điểm mà nguyên tắc mang lại?

2.2 Quy trình chế tạo CMOS

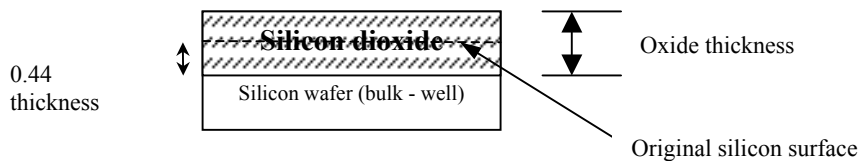


Hình 2-6 Chế tạo IC. Phát triển crystalline silicon (1); tạo wafer (2–3); oxidation - tạo lớp silicon dioxide (oxide) trong lò luyện (4); ứng dụng chất cản quang (5); ứng dụng lớp photomask để làm cứng lớp cản quang (6); wafer với lớp cản quang mềm đã bị loại bỏ (7); khắc axit lớp oxide (8); cấy ion (9–10); gỡ bỏ hoàn toàn lớp cản quang (11); gỡ bỏ lớp oxide (12).

Các bước cơ bản chế tạo IC công nghệ bán dẫn CMOS bao gồm:

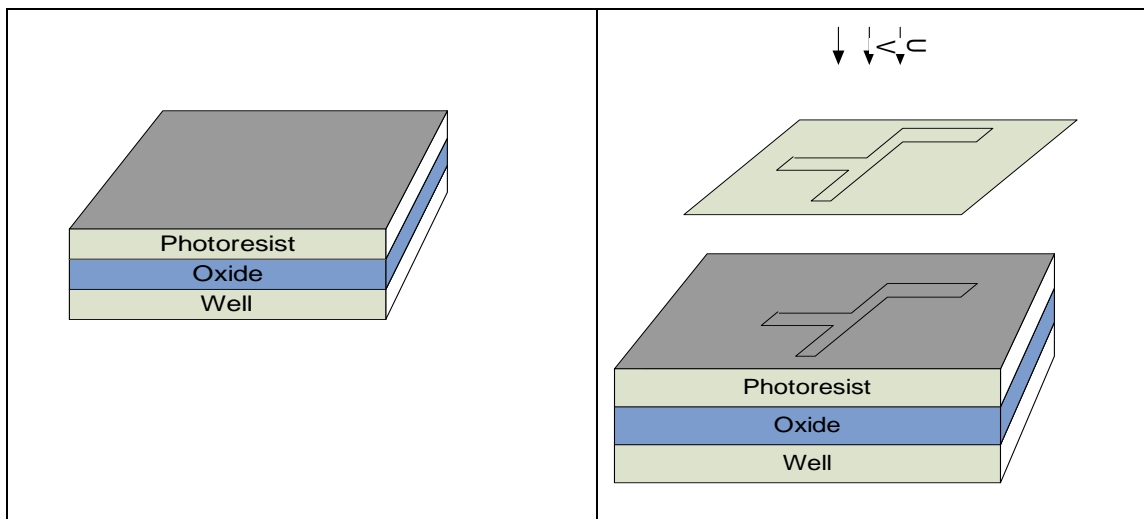
1. **Silicon Wafer:** Silicon được tinh chế từ thạch anh (nhỏ hơn 1 tạp chất trên 10^{10} nguyên tử silicon). Đặt thỏi silicon tinh thể đơn trong lò nung có điểm nóng chảy ở 1500°C (điểm nóng chảy của silicon tại áp suất 1 atm là 1414°C). Các chất kích tạp loại p (chất nhận) hay n (chất cho) có thể được thêm vào để phát triển loại silicon mong muốn. Công nghệ CMOS dưới 1um hay dùng silicon wafer loại p. Dùng cưa kim cương cắt thỏi silicon thành các bánh wafer đường kính 6-12inches, dày 600um. Các góc vác chỉ định hướng tinh thể.
2. **Phủ oxide - Oxidation:** Để tiếp tục việc chế tạo IC, thoát đầu cho các bánh wafer xếp kề nhau trong lò nung để phủ lên bề mặt wafer lớp silicon dioxide

(oxide). Oxide phát triển lên trên bề mặt song cũng đồng thời thâm vào trong wafer. Thông thường, oxide thickness từ 150-10000 Å

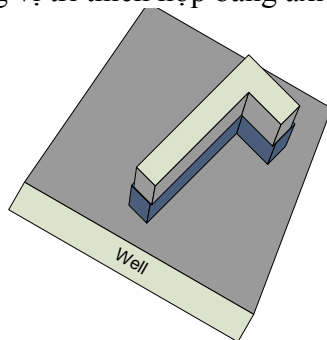


Hình 2-7 Phát triển silicon dioxide trên bề mặt silicon wafer

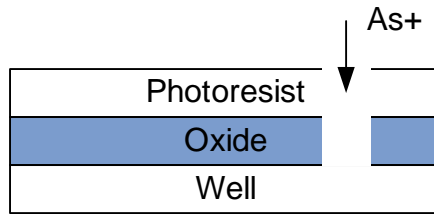
3. **Tạo lớp cản quang photoresist – Deposition (vẽ hình):** tạo lớp cản quang (có thể là âm hay dương - either negative or positive photoresist) trên bề mặt wafer. Chiếu tia cực tím xuyên qua lớp photomask nhằm làm thay đổi độ cứng lớp cản. Lớp cản quang được làm cứng có nhiệm vụ bảo vệ lớp poly bên dưới trong quá trình khắc axit sau đó.



4. **Khắc axit hoặc plasma - Etching (vẽ hình):** loại bỏ lớp cản quang và polysilicon tại những vị trí thích hợp bằng axit hoặc plasma.



5. **Khuếch tán – Diffusion hoặc Cấy ion - Ion Implantation:** những ion của chất kích tạp loại n hay p được gia tốc với vận tốc lớn vào tấm silicon wafer (bulk) với chiều sâu từ 0.1 - 0.6μm. Khuếch tán hay cấy ion đều có cùng mục đích, song cấy ion có nhiều ưu điểm hơn.

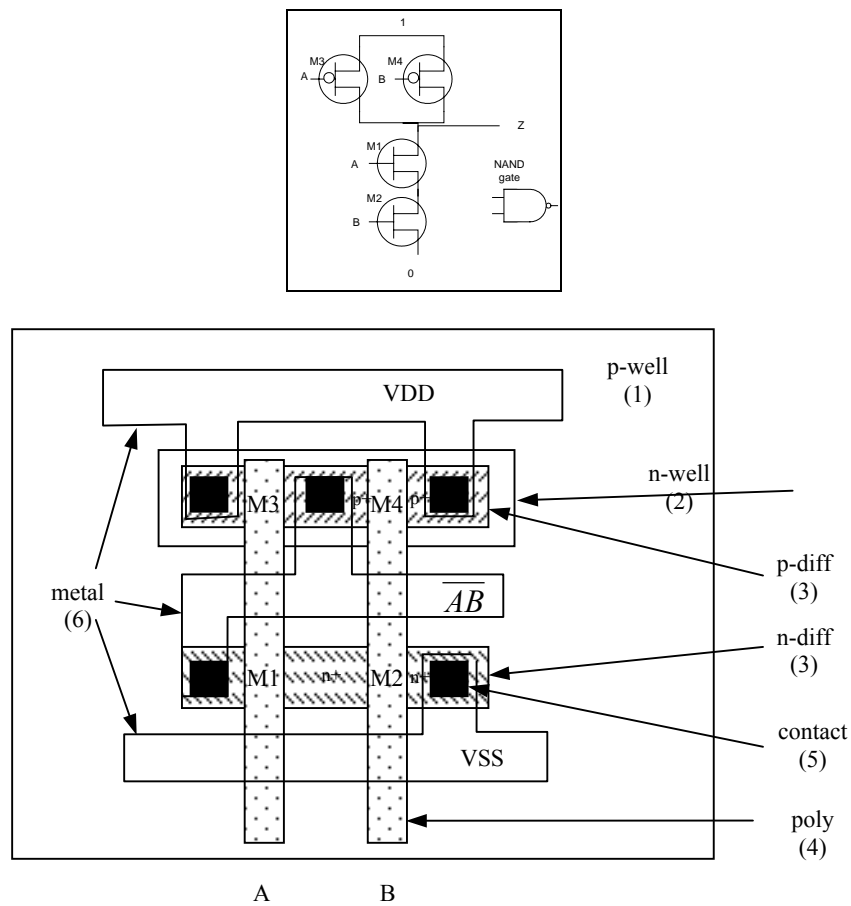


6. Loại bỏ lớp cản quang.

2.3 Quy luật thiết kế

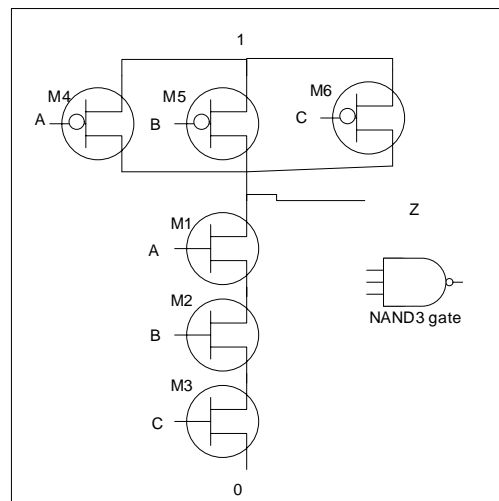
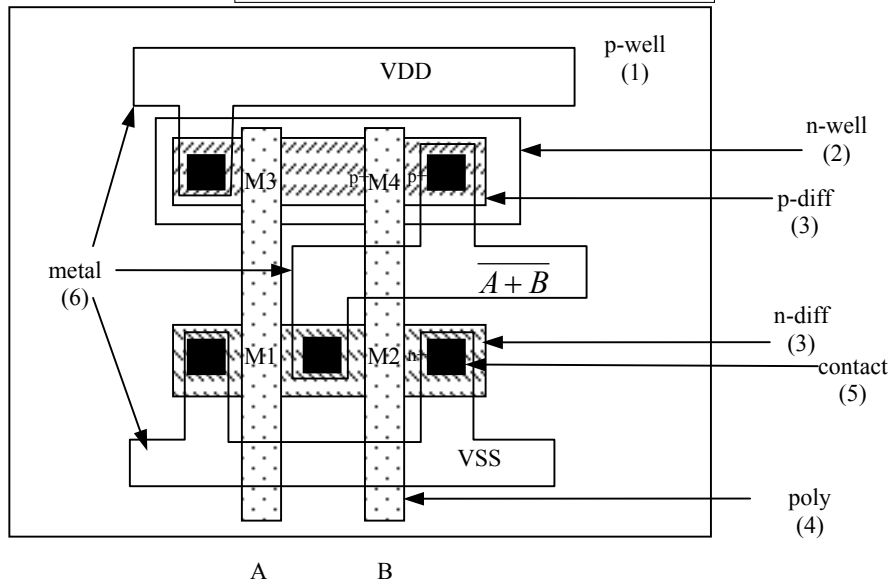
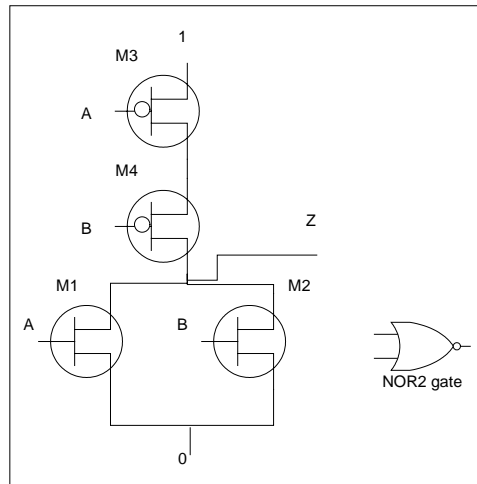
SV tự đọc sách. Further in [Neil & Kamran, Principle of CMOS VLSI Design – section 3.4]

Layout cho cổng CMOS NAND2:

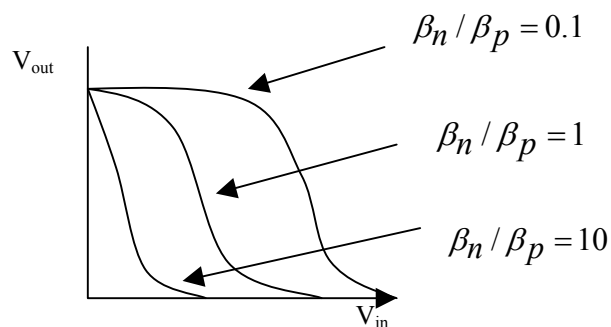


Thiết kế layout cho clocked-inverter?

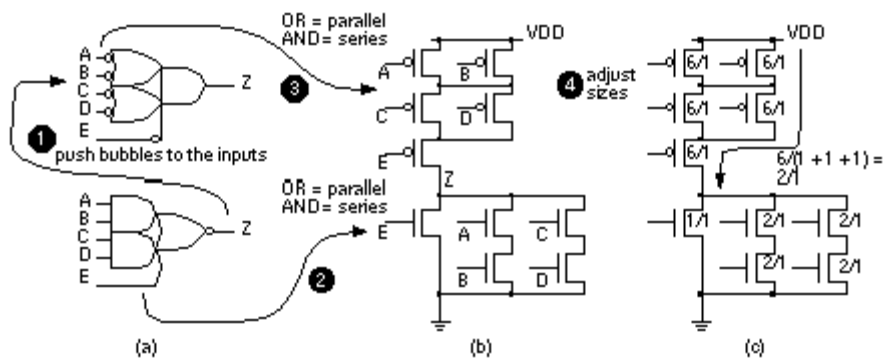
Vẽ layout cho NOR2, NAND3?



Mục đích sâu xa của **độ bền điều khiển**: là vì $I_{DS} \sim \beta_n = k'_n \frac{W}{L}$ cho nên $\beta_n / \beta_p = 1$ cho phép tụ tải (capacitive load) nạp và xả với khoảng thời gian bằng nhau, nghĩa là dòng do INV cung cấp và thu nhận là như nhau.



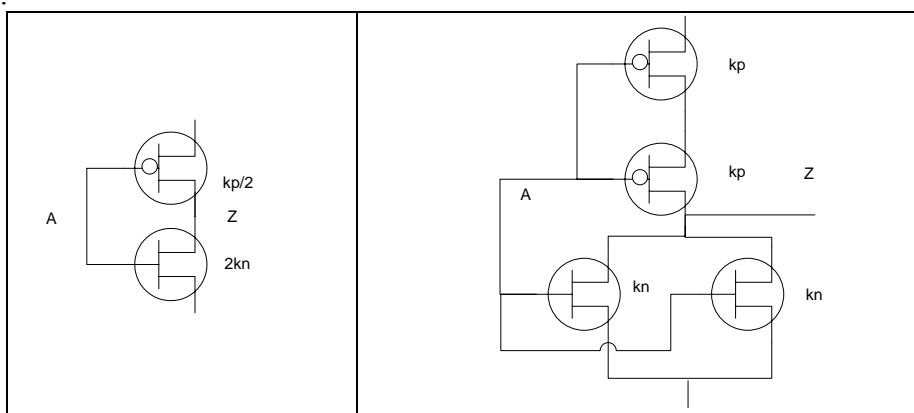
Bài toán trong hình vẽ cho KQ là 2/1 cho phần nMOS và 1/1 cho phần pMOS.



Hình 2-9 AOI221

Giả sử các transistor có kích thước cố định và bằng nhau. INV cấu thành bởi 2 transistor có **tham số hỗ dẫn** lần lượt là $2k_n$ và $k_p/2$ ($k_p = 4k_n$). Nếu muốn dùng NOR2 để tạo INV đạt được **độ bền điều khiển** thì các transistor trong NOR2 phải có **tham số hỗ dẫn** là bao nhiêu? Giải thích, vẽ hình?

Trả lời:



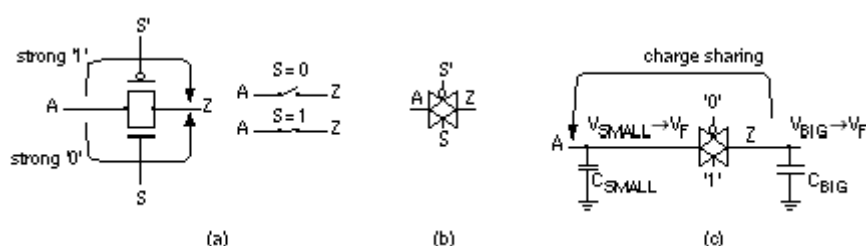
INV với các giả thiết đã cho là đạt được độ bền điều khiển. Đã biết: R tỷ lệ nghịch với $\beta = k \frac{W}{L}$, trong đó $\frac{W}{L}$ cố định. Cho nên:

$$\frac{1}{k_{pINV} / 2} = R_{pINV} = R_{pNOR2} + R_{pNOR2} \Rightarrow \frac{2}{k_{pNOR2}} \Rightarrow k_{pNOR2} = k_{pINV}$$

$$\frac{1}{2k_{nINV}} = R_{nINV} = R_{nNOR2} // R_{nNOR2} \Rightarrow \frac{1}{2k_{nNOR2}} \Rightarrow k_{nNOR2} = k_{nINV}$$

Tính toán con số 224 cells (for AOI family)

2.4.3. TG & MUX



Hình 2-10 CMOS transmission gate (TG). (a) transistor kênh p và n mắc song song tạo thành TG. (b) Ký hiệu TG thông dụng (c) hiện tượng chia sẻ điện tích

Biểu thức TG: $Z = TG(A, S)$

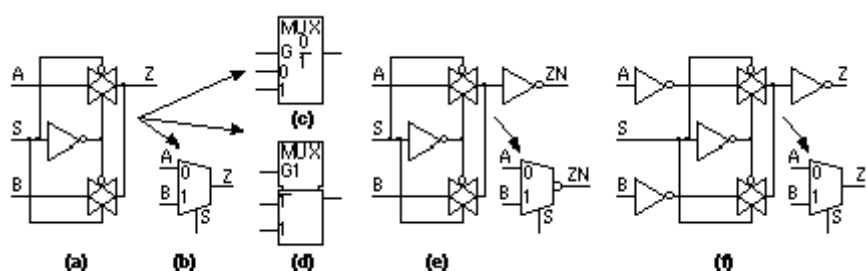
Chỉ ra ưu điểm sử dụng 2 transistor s/v pass transistor, vẽ, giải thích.

Nhược của TG: hiện tượng chia sẻ điện tích (charge sharing) với phần mạch như trên – cho ví dụ chứng minh. $V_F = \frac{V_{SMALL}C_{SMALL} + V_{BIG}C_{BIG}}{C_{SMALL} + C_{BIG}}$. $C_{BIG} = 0.2\text{pF}$ (10 lần giá trị tải chuẩn với G5) $C_{SMALL} = 0.02\text{pF}$. $V_{BIG} = 0\text{V}$, $V_{SMALL} = 5\text{V}$ thì tính được $V_F = 0.45\text{V}$.
Cách cải tiến: ngõ vào đủ mạnh hoặc là dùng buffer giữa A và Z

Thiết kế MUX: 2 cách:

1. sử dụng TG
2. sử dụng tế bào logic AOI, OAI

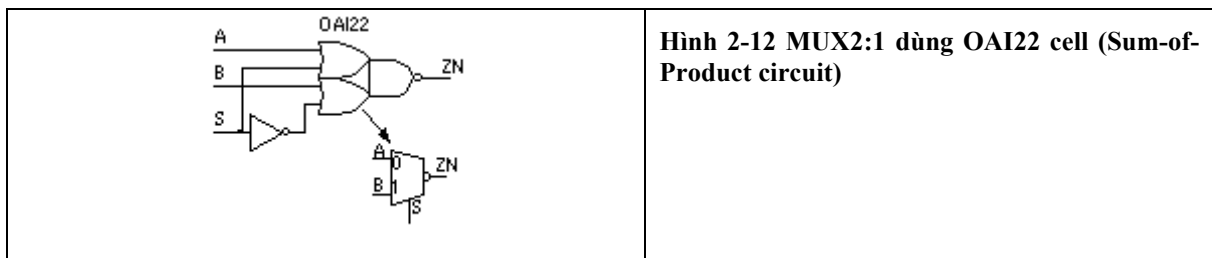
Cách 1: Thiết kế MUX2:1 dùng TG: $Z = TG(A, S') + TG(B, S) = A.S' + B.S = \text{MUX}(A, B, S)$



Hình 2-11 CMOS MUX. (a) MUX2:1 dùng TG không có bộ đệm (b) Ký hiệu MUX (c) Ký hiệu MUX theo chuẩn IEEE (d) Ký hiệu MUX phổ biến (theo IEEE) (e) MUX đảo có đệm ngõ ra (f) MUX2:1 có đệm vào ra.

Giải thích. Các trường hợp (e) và (f) là cải tiến của (a). Giải thích ý nghĩa G và cách thức tác động. Nếu là MUX4:1 ta có w0,1,2,3 với s1,0.

Cách 2: Thiết kế MUX2:1 đảo dùng OAI cell: $Z_N = A'S' + B'S = [(A'S)'] \cdot (B'S)']' = [(A+S)(B+S')]']' = \text{OAI22}[A,S,B,\text{NOT}(S)]$



Thiết kế MUX2:1 sử dụng tế bào logic AOI?

Thiết kế MUX 4:1 dùng AOI cell?

Thiết kế MUX 4:1 dùng MUX 2:1?

Thiết kế MUX 16:1 dùng MUX 4:1?

Thiết kế tri-state buffer dùng TG tránh hiện tượng chia sẻ điện tích?

XOR

$$\text{XOR}(A1,A2) = A1.A2' + A1'.A2 = \text{MUX}(A1, \text{NOT}(A1), A2)$$

$$\text{XOR}(A1,A2) = \text{NOT}[\text{MUX}(\text{NOT}(A1), \text{NOT}(\text{NOT}(A1)), A2)]$$

$$\text{XOR}(A1,A2) = A1.A2' + A1'.A2 = [A1.A2 + A1'.A2']' = [(A1.A2) + (A1+A2)']' = [(A1.A2) + \text{NOR}(A1,A2)]' = \text{AOI21}[A1,A2,\text{NOR}(A1,A2)]$$

XNOR

$$\text{XNOR}(A1,A2) = A1.A2 + A1'.A2' = \text{NOT}[\text{NOT}[\text{MUX}(A1, \text{NOT}(A1), A2)]] = \text{OAI21}[A1,A2,\text{NAND}(A1,A2)]$$

Chỉ ra các ưu điểm TG so với pass transistor?

Thiết kế XOR2 sử dụng TG sao cho tổng số transistor là 8 hoặc 6?

2.5 Tế bào logic tuần tự (Sequential Logic Cell)

So sánh mạch logic tổ hợp (CL) và logic tuần tự (SL).

Moore & Mealy.

FSM.

Hai pp clocking chính trong công nghệ VLSI:

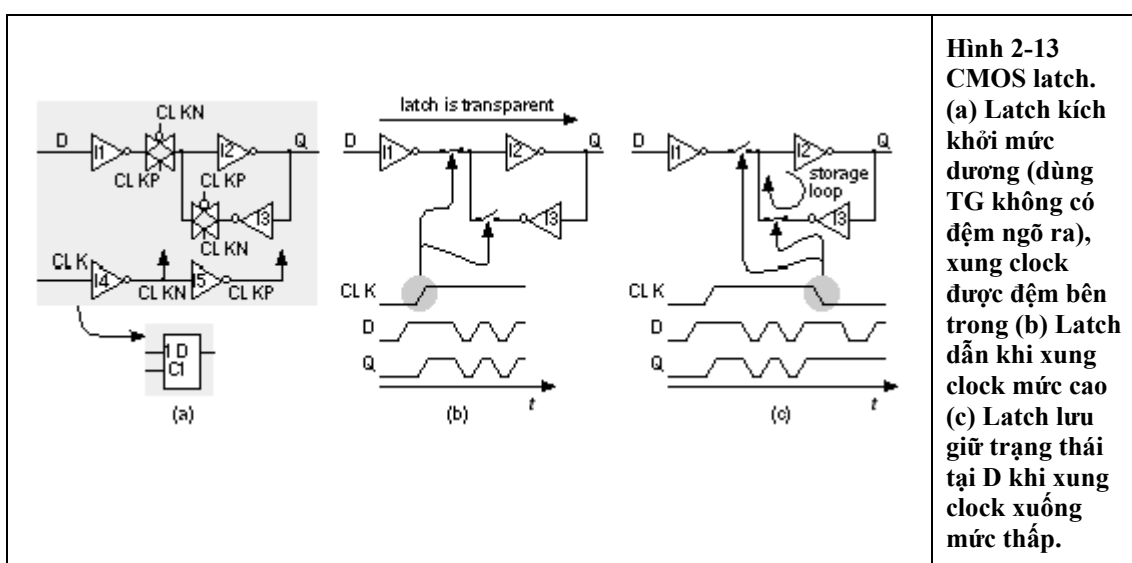
1. Xung đồng hồ đa pha hoặc đơn pha - multiphase or single clock
2. Thiết kế đồng bộ - synchronous design

Cách 2 có nhiều ưu điểm hơn như

1. Cho phép thiết kế tự động
2. An toàn
3. Cho phép thực thi ASIC giống như đã mô phỏng - vendor signoff

2.5.1. Bộ chốt dữ liệu – latch or D-latch

Hình dưới đây chỉ ra 1 logic cell tuần tự - latch hay D-Latch. Xung đồng hồ bên trong (nội) CLKN (N for negative) & CLKP (P for positive), tạo từ xung clock hệ thống CLK, bởi 2 cổng đảo (I4, I5). Hai cổng đảo này là bên trong latch. Tuy có thể tiết kiệm không gian song sẽ nguy hiểm nếu tạo các tín hiệu này bên ngoài.



Để nhấn mạnh sự khác nhau giữa latch với FF người ta gọi ngõ vào clock là tín hiệu cho phép (enable), nhìn vào hình (b), khi xung clock mức cao, Latch dẫn thông, nghĩa là mọi thay đổi ở D dẫn đến sự thay đổi ngõ ra Q (rất khác so với FF sẽ xem xét sau). Còn khi clock xuống mức thấp, như trong hình (c), cổng đảo I2 và I3 kết nối với nhau tạo nên vòng lưu giữ trạng thái cũ tại D cho đến khi có clock mức cao trở lại. Vòng này thực hiện công việc lưu giữ chừng nào còn cung cấp nguồn, cho nên gọi đây là latch tĩnh. Logic tuần tự khác với logic tổ hợp là vì đặc điểm lưu trữ hay nhớ này (feature of storage or memory).

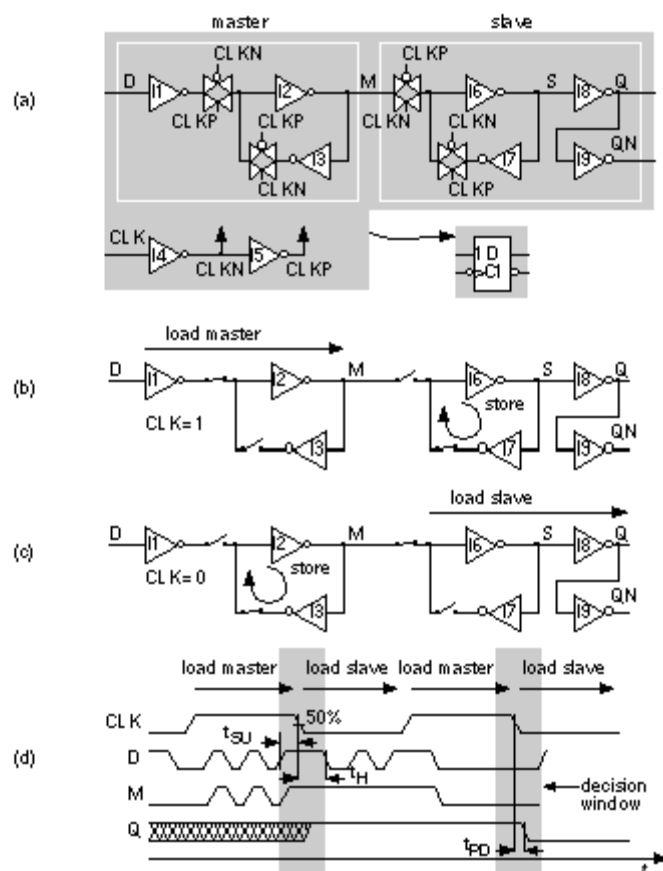
Ngõ ra Q là không đệm và kết nối trực tiếp đến ngõ ra I2, chính là nút lưu trữ. Thư viện ASIC thường có thêm 2 INV ngõ ra cho Q và QN nhằm đảm bảo cách ly cho nút lưu trữ. Khi này Latch bao gồm 7 INV và 2 TG bên trong (4.5 gates).

Latch kích khởi mức âm: thêm cổng đảo cho các xung clock bên trong hoặc hoán đổi chức năng CLKN với CLKP.

2.5.2. Flip-Flop

Dùng 2 D-latch (master latch & slave latch) xây dựng Flip-flop như trong hình. FF gồm 9 inverters & 4 TGs, tức tổng cộng 6.5 gates. Nút lưu trữ S được đệm.

Trễ clock-to-Q + trễ inverter = trễ clock-to-QN.



Hình 2-14 CMOS flip-flop. (a) FF kích khởi cạnh âm gồm latch chủ và tớ (b) Khi clock mức cao, latch chủ dẫn thông (c) Khi clock mức thấp, latch tớ chuyển tải giá trị của latch chủ. (d) Dạng sóng mô tả định nghĩa setup time t_{SU} , hold time t_H , & clock to Q propagation delay t_{PD} của FF.

Khi clock mức cao: latch chủ dẫn thông, nút M biến đổi theo ngõ vào D. Latch tớ bị tách biệt với latch chủ và lưu trữ trạng thái M trước đó.

Khi clock chuyển xuống mức thấp: latch tớ dẫn thông khiến cho trạng thái ngõ ra Q được cập nhật theo trạng thái nút M. Giá trị của M ngay tại cạnh âm xung clock sẽ được lưu giữ tại ngõ ra Q bất chấp thay đổi tại ngõ vào D khi clock ở mức thấp. Khi clock lên mức cao trở lại, latch tớ giữ lại giá trị M cũ trên. Tiến trình cứ thế tiếp tục.

Kết hợp 2 latch như trên để có thể lấy mẫu ngõ vào D tại cạnh âm xung clock, tức ta có FF kích khởi cạnh âm. Rõ ràng hành vi FF rất khác với latch.

Hình (d): giữ cho dữ liệu ổn định (logic '1' hay '0') khoảng thời gian t_{SU} trước cạnh âm xung clock và t_H sau cạnh âm xung clock. Điểm tham chiếu là giữa cạnh âm (50%VDD) – trip point.

D-FF trên được dùng rất phổ biến trong thiết kế ASIC. Một số loại khác cũng được dùng trong ASIC cell lib là JK-FF, T-FF (toggle), & SR-FF nhằm tương thích với các thiết kế TTL. Đôi khi khái niệm register để chỉ 1 tập các FFs hoặc latch, song nó cũng

có nghĩa là chỉ 1 FF hay latch mà thôi. Tác giả ngầm định register có hơn 1 FF hay latch.

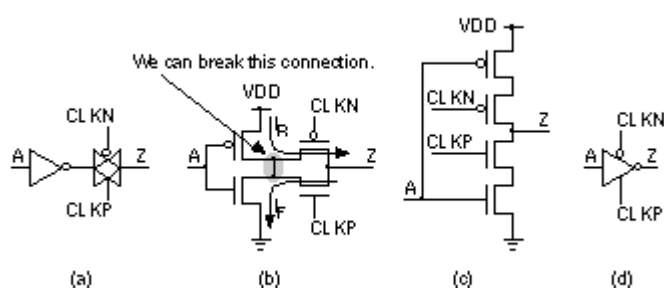
Để tạo DFF với các chân set, reset, ta thể INV (cả trong latch chủ và tớ) bởi NAND2 cell. Set tác động mức thấp: thế I2 và I7. Reset tác động mức thấp: thế I3 và I6.

Chỉ thế 1 INV được không?

Vài loại TTL FF có reset hoặc set trội (dominant), nhưng rất khó làm điều này trong ASIC. Set đôi khi được gọi là preset (IEEE ký hiệu là 'P'). Reset đôi khi gọi là clear ('R').

2.5.3. Cổng đảo có xung clock - Clocked Inverter

Hình vẽ chỉ rõ thiết kế clocked inverter dùng inverter và TG. Mũi tên chỉ ra chiều dòng điện khi nạp (I_R) và xả (I_F) của tụ tải qua TG. Dễ thấy rằng ta có thể gỡ bỏ mối nối mà không làm thay đổi bản chất hoạt động của mạch. Ký hiệu trong hình tuy thông dụng song không chuẩn.



Hình 2-15 Clocked inverter. (a) inverter + transmission gate (TG). (b) Dòng nạp xả tụ tải qua TG. Việc gỡ bỏ mối nối không làm ảnh hưởng hoạt động mạch. (c) gỡ bỏ mối nối tạo thành clocked inverter. (d) ký hiệu thông dụng

Ta có thể sử dụng clocked inverter thay thế cho cặp INV-TG trong latch hay FF.

Latch: thay thế I1 & I3 (cùng với TGs theo sau) bởi clocked inverter. Trong TH của latch hơi khó nhận thấy ưu điểm nào vượt trội, ngoại trừ việc thiết kế layout dễ dàng hơn vì có ít hơn 1 mối nối.

FF: thay thế I1, I3 & I7 (cùng với TGs theo sau) bởi clocked inverters. Còn việc thế I6 là không được. Thế TG sau M bởi clocked inverter sẽ làm thay đổi Q ngõ ra thành QN. Khi này: Trễ clock-to-QN + trễ inverter = trễ clock-to-Q.

Theo đó, muốn trễ clock-to-QN ngắn thì sử dụng clocked inverters, ngược lại, cần trễ clock-to-Q ngắn thì dùng INV-TG. Thực tế, ta không dùng cả Q và QN cho nên vài thư viện chỉ có Q hoặc QN để tiết kiệm tài nguyên. Layout cho clocked inverter dễ xây dựng hơn INV-TG, vì vậy FF trong thương mại hay sử dụng cả 2 cách thực thi.

2.6 I/O cell

(Xem sách)

2.7 Trình dịch cell - Cell Compiler

(Xem sách)

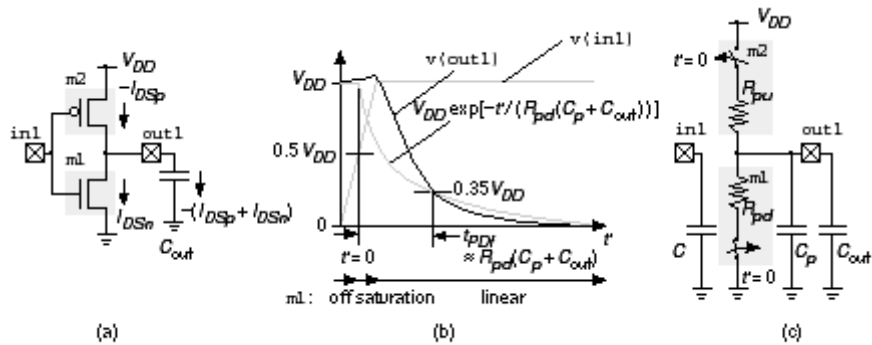
CHƯƠNG 3: Thiết kế thư viện ASIC

Drive strength: 1X, 2X.

Hệ số scale s.

Tỷ lệ logic r.

3.1 Mô hình trễ của transistor



Hình 3-1 Mô hình trễ logic (a) CMOS INV với tải C_{out} (b) falling propagation delay, t_{PDF} (input trip point: 0.5, output trip point: 0.35 (falling) and 0.65 (rising)). Mô hình cho ra $t_{PDF} \approx R_{pd}(C_p + C_{out})$. (c) Mô hình cho INV bao gồm: tụ ngõ vào, C ; điện trở kéo lên R_{pu} & điện trở kéo xuống R_{pd} ; tụ ngõ ra ký sinh C_p

Bài toán: Vin thay đổi từ 0 đến VDD, ngõ ra Vout thay đổi từ VDD đến 0 (falling). Tính thời gian truyền lan t_{PD} với các trip point như đã cho?

Thật vậy, ban đầu M1 tắt, vin tăng kéo theo M1 dẫn trong miền bão hòa, sau đó dẫn trong miền tuyến tính. Ta mô hình M1 bởi R_{pd} (kéo xuống), M2 bởi R_{pu} (kéo lên).

Trễ là do R_{pd} , R_{pu} , C_p , C_{out} tạo nên. Nếu giả sử $R_{pd} = \text{const}$ thì vout đạt tới giá trị $0,35V_{DD}$ khi :

$$0,35V_{DD} = V_{DD} \exp\left[\frac{-t_{PDF}}{R_{pd}(C_{out} + C_p)}\right]$$

Chọn 0,35 là vì $\ln \frac{1}{0,35} \approx 1$ nên $t_{PDF} \approx R_{pd}(C_{out} + C_p)$

3.2 Tụ ký sinh

(Xem sách)

3.3 Logical Effort

Gọi tq là trễ do sự không lý tưởng gây nên, ta có tq bao gồm:

- Trễ phụ thuộc tụ ký sinh bên trong
- Trễ phụ thuộc thời gian V_{in} đạt tới V_{th} (ngưỡng) của cell
- Trễ phụ thuộc slewrate của ngõ vào

Thì $t_{PD} \approx R (C_{out} + C_p) + t_q$

Với C5 (Compass, 3,3V, 0,5um) (Compass Lib sử dụng mô hình PSPICE phức tạp và chính xác hơn G5), thì với NAND2 cell 1X drive, ta có :

$$t_{PD} \approx (0,07 + 1,46C_{out} + 0,15)ns$$

Số hạng thứ nhất ứng với RC_p , số hạng thứ hai tức $R \sim 1,46K$, số hạng ba là t_q .

Trễ cho bài toán rising cũng xấp xỉ như cho bài toán falling vừa đạt được.

Nếu cell có hệ số scale s (W lớn gấp s lần, $L = \text{const}$), thì

$$R \rightarrow R/s$$

$$C_p \rightarrow sC_p$$

t_q không lý tưởng cho nên khó tiên đoán (chỉ giả sử là thay đổi tuyến tính)

Vậy, t_{PD} cho cell có hệ số scale s là $t_{PD} \approx \frac{R}{s}(C_{out} + sC_p) + st_q$

Ví dụ, NAND2 cell 2X drive, $s=2$ thì $t_{PD} \approx (0,03 + 0,75C_{out} + 0,51)ns$, theo đó, R_p và C_p là đúng tiên đoán, trong khi t_q rất khác hoàn toàn với giả thiết đặt ra ở trên.

Viết lại $t_{PD} \approx \frac{R}{s}(C_{out} + sC_p) + st_q$ sử dụng C_{in} của cell có hệ số scale s là $C_{in} = sC$, ta

$$\text{được } t_{PD} \approx RC \frac{C_{out}}{C_{in}} + RC_p + st_q$$

Đơn vị hóa t_{PD} bằng cách chia cho $\tau = R_{inv}C_{inv}$. Với R_{inv} là điện trở kéo, C_{inv} tụ ngõ vào của INV 1X (minimum).

$$d = \frac{t_{PD}}{\tau} = \frac{RC \frac{C_{out}}{C_{in}} + RC_p + st_q}{\tau} = f + p + q$$

τ được xem là đặc tính cơ bản của công nghệ CMOS.

C5, INV 1X drive $t_{PD} = (0,06 + 1,6C_{out} + 0,1)ns$ tức $R_{inv} = R_{pd} = R = 1,6K$

C5, $\tau = R_{inv}C_{inv} = (1,6K).(0,036pF) \approx 0,06ns$ (C_{inv} from data book)

$$d = f + p + q$$

= **effort delay + parasitic delay + nonideal delay**

$$= \mathbf{g * h + p + q}$$

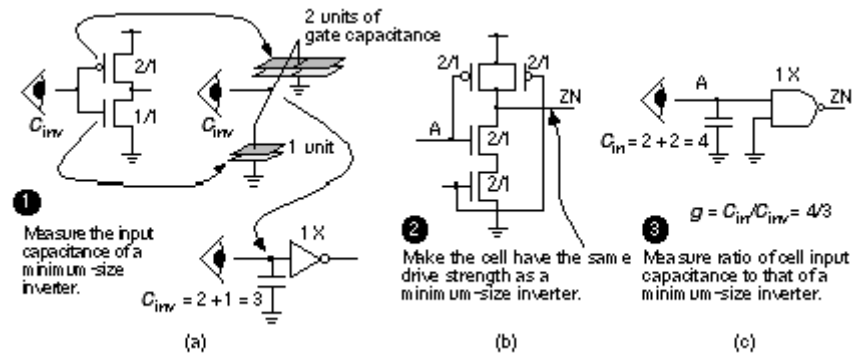
= **logical effort * electrical effort + parasitic delay + nonideal delay**

Logical effort g: chỉ phụ thuộc loại cell mà không phụ thuộc kích cỡ cell.

$$g = \frac{RC}{\tau} = \frac{\text{const}}{\tau} \text{ vì } R \rightarrow R/s, C \rightarrow sC$$

Cách tìm g: thay đổi cell sao cho có cùng drive (độ bền điều khiển) như INV 1X, thì

$$g = \frac{C_{in}}{C_{inv}}, \text{ với } C_{in} \text{ là của cell vừa thay đổi.}$$



Hình 3-2 Logical effort (a) C_{in} của INV 1X (min size) (b) thay đổi cell sao cho cùng độ bền điều khiển như INV 1X (hệ số logic bằng 2), tính C_{in} (c) tìm g , $g=4/3$

Logical effort phụ thuộc hệ số logic r . Xem bảng.

Electrical effort h : phụ thuộc C_{out} (tụ tải) & C_{in} (tụ ngõ vào) của logic cell

$$h = \frac{C_{out}}{C_{in}}$$

Parasitic delay p : $p = \frac{RC_p}{\tau}$

Cell	Cell effort (logic ratio = 2)	Cell effort (logic ratio = r)	Parasitic delay/ τ	Nonideal delay/ τ
inverter	1 (by definition)	1 (by definition)	p_{inv} (by definition) <u>1</u>	q_{inv} (by definition) ¹
n -input NAND	$(n + 2)/3$	$(n + r)/(r + 1)$	$n p_{inv}$	$n q_{inv}$
n -input NOR	$(2n + 1)/3$	$(nr + 1)/(r + 1)$	$n p_{inv}$	$n q_{inv}$
$C5, p_{inv}=1, q_{inv}=1,7, R_{inv}=1,5K, C_{inv}=0,036pF$				

Table 3-1 Cell effort, parasitic delay, nonideal delay (in units of τ) cho cell CMOS đơn tầng

Ý nghĩa chính của logical effort?

3.3.1. Ước tính trễ

Tính delay cho NOR3 2X drive điều khiển 4 ngõ ra (fanout of 4) và tải 0.3 pF (bao gồm tụ input của 4 cell và tụ interconnect). Từ bảng trên, ta có $p = 3 p_{inv}$ và $q = 3 q_{inv}$. C_{in} cho NOR3 1X là $g C_{inv}$, do đó cho NOR3 2X cell, $C_{in} = 2 g C_{inv}$. Do vậy

$$gh = g \frac{C_{out}}{C_{in}} = \frac{g \cdot (0.3 \text{ pF})}{2 g C_{inv}} = \frac{(0.3 \text{ pF})}{(2) \cdot (0.036 \text{ pF})} \quad (3.27)$$

(Nhận thấy **g không ảnh hưởng đến delay**, ta sẽ xem xét tiếp trong phần 3.3.3)

Delay của NOR logic cell tính theo đơn vị τ (tau), là

$$0.3 * 10^{-12}$$

$$d = gh + p + q = \frac{\quad}{(2) \cdot (0.036 * 10^{-12})} + (3) \cdot (1) + (3) \cdot (1.7)$$

$$= 4.1666667 + 3 + 5.1$$

$$= 12.266667 \tau . \quad (3.28)$$

Tương ứng delay tuyệt đối là $t_{PD} = 12.3 * 0.06 \text{ ns} = \mathbf{0.74 \text{ ns}}$.

Delay cho 2X drive, NOR3 logic cell trong thư viện C5 là

$$t_{PD} = (0.03 + 0.72 C_{\text{out}} + 0.60) \text{ ns} . \quad (3.29)$$

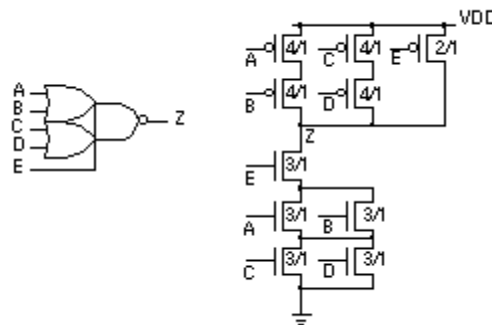
Với $C_{\text{out}} = 0.3 \text{ pF}$,

$$t_{PD} = 0.03 + (0.72) \cdot (0.3) + 0.60 = \mathbf{0.846 \text{ ns}} . \quad (3.30)$$

Kết luận: So với giá trị ước đoán của chúng ta là 0.74 ns. Sai lệch ở đây chủ yếu do sai lệch trong ước đoán nonideal delay. Logical effort cho ta pp khảo sát các delay tương đối song không hoàn toàn chính xác. Song quan trọng hơn là logical effort cho chúng ta hiểu vì sao logic gây ra delay.

3.3.2. Diện tích logic & hiệu quả logic

Hình dưới chỉ ra OAI cell đơn tầng: có các logical effort khác nhau tại các input. Logical effort cho OAI221 là **logical-effort vector** $g = (7/3, 7/3, 5/3)$. VD, thành phần đầu tiên của vector 7/3 chính là logical effort của input A & B trùng hình.



Hình 3-3 OAI221 $g = (7/3, 7/3, 5/3)$. Logical effort cho input A và B là 7/3, logical effort cho input C và D là 7/3, cho input E là 5/3. Diện tích logic là 33 logical square

Gọi diện tích transistor kênh n 1X (min) là **logical squares**. Tính diện tích các transistor trong logic cell (bỏ qua routing area, drain area, & source area) theo logical square. Diện tích logic của OAI221 1X drive tính như sau:

- n -channel transistor sizes: $3/1 + 4 * (3/1)$
- p -channel transistor sizes: $2/1 + 4 * (4/1)$
- diện tích logic tổng cộng = $2 + (4 * 4) + (5 * 3) = 33 \text{ logical squares}$

AOI221 cell đơn tầng trong hình sau, với $g = (8/3, 8/3, 6/3)$. Tương tự:

- n -channel transistor sizes: $1/1 + 4 * (2/1)$

- p -channel transistor sizes: $6/1 + 4 * (6/1)$
- diện tích logic tổng cộng = $1 + (4 * 2) + (5 * 6) = 39$ logical squares

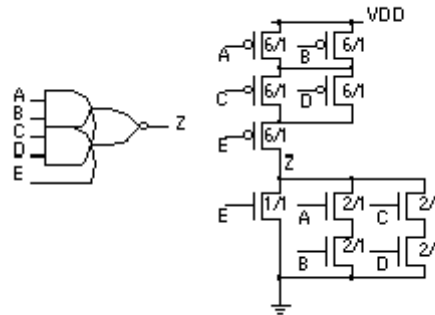


Figure 3-4 AOI221, với logical-effort vector, $\mathbf{g} = (8/3, 8/3, 7/3)$. Diện tích logic là 39 logical squares.

Nhận xét: OAI221 đơn tầng với diện tích logic = 33 logical squares và logical effort $(7/3, 7/3, 5/3)$ cho hiệu quả logic cao hơn AOI221 đơn tầng với 39 logical squares và logical effort cũng lớn hơn $(8/3, 8/3, 6/3)$.

3.4 Bài tập

Pull resistance. Chỉ ra rằng với V_{DS} nhỏ, n-transistor giống như 1 điện trở

$$R = \frac{1}{\beta_n (V_{DD} - V_{tn})} \cdot \text{Với } V_{DS} = 0, \text{ tính } R \text{ cho transistor } \frac{6}{0,6} \text{ trong miền tuyến}$$

$$k_n = 200 \mu A V^{-2}$$

tính?

Tính logical effort vector cho AOI221 cell với logical ratio r thay đổi?

(XOR cell & logical effort). Chỉ ra cách thực hiện XOR2 sử dụng AOI22 & 2 INV. Dựa vào logical effort để so sánh cách xây dựng này với trường hợp sử dụng AOI21 & 1 NOR ?

Bài 3.11 (*AOI & OAI cell efficiency) 1 thư viện cell chuẩn có các dữ liệu sau:

AOI221: $t_R = 1.06-1.15\text{ns}$; $t_F = 1.09-1.55\text{ns}$; $C_{in} = 0.21-0.28\text{pF}$; $W_C = 28.8 \mu\text{m}$

OAI221: $t_R = 0.77-1.05\text{ns}$; $t_F = 0.81-0.96\text{ns}$; $C_{in} = 0.25-0.39\text{pF}$; $W_C = 22.4 \mu\text{m}$

(W_C : cell width; cell height $25.6 \mu\text{m}$.)

(a) Tính logical effort

(b) Tính diện tích logic cho AOI221 và OAI221 cells. Thực thi OAI221 kiểu đơn tầng: $\text{OAI221} = \text{OAI221}(a1, a2, b1, b2, c)$, Thực thi AOI221 kiểu đa tầng: $\text{AOI221} = \text{NOT}(\text{NAND}(\text{NAND}(a1, a2), \text{AOI21}(b1, b2, c)))$.

(c) Có cách thực thi nào khác cho 2 cell này không?

(d) Giải thích sự thực thi đó

Bài 3.18 (Set & reset, 10 phút.) Chỉ ra cách thêm synchronous set hoặc synchronous reset vào FF trong hình 2.18(a) sử dụng 2:1MUX.?

CHƯƠNG 4: VHDL (12 tiết)

LED 7 đoạn

Bộ đếm.

Bộ điều khiển đèn giao thông.

XILINX FPGA KIT.