[Iain McNally]

Components for Logic

Diode

Bipolar Transistors

MOS Transistors

RTL Inverter and NOR gate

Inverter is a special case of a NOR gate with only one input

RTL → NOR ×2
TTL → NAND ×3
NMOS → NOR + NAND
CMOS → Compound Gate ×2

NMOS

$Z = \overline{A \cdot B}$

Digital CMOS Circuits

RTL → One input → One Transistor
TTL → 同上
NMOS → NOT → ×2 NOR ×3
CMOS → Input ×2

All CMOS devices are enhancement mode

Compound Gates

$Z = \overline{(A \cdot B) + C}$

$\bar{Z} = (A \cdot B) + C$ (pull down)

$Z = (\bar{A} + \bar{B}) \cdot \bar{C}$ (pull up)

All compound gates are inverting
Compound gate要连着画!!!

Pull up circuit 可以用 NMOS 替代

A logical approach to gate layout

Euler Path: Find an arrangement of transistors such that we can butt adjoining transistors
- Careful selection of transistor ordering
- Careful orientation of transistor source and drain

Referred to as line of diffusion.

Finding an Euler Path

① Find a path which passes through all n-transistors exactly once
② Express the path in term of the gate connections

Multiple gates

Gates should all be of same height
- Power and ground rails will then line up when butted.

All gate inputs and outputs are available at top and bottom.
- All routing is external to cells
- Preserves the benefits of hierarchy

Interconnect is via two conductor routing
- In this case Polysilicon vertically and Metal horizontally.

1 Metal Layer

2 Metal Layer

P-substrate

Polysilicon

Active Area Mask

N-type implant has no extra mask
It is blocked by thick oxide and by polysilicon
The implant is self Aligned
substrate connection is to bottom of wafer
All substrates to ground
Active Area mask defines extent of Thick Oxide
Polysilicon mask also controls extent of Thin Oxide

Resistance

resistivity constant

$R = (\frac{\rho}{t}) \cdot (\frac{L}{W})$

$\frac{\rho}{t}$ fixed

$R = R_S \cdot (\frac{L}{W})$

NMOS Transistor - with top substrate connection

P well mask

P+ implant mask

N+ implant task

Processing - Photolithography

Contact/Proximity Mask

Glass mask

Mask pattern (chrome)

Silicon Wafer

Oxide Growth SiO_2

Photoresist Deposition

Photoresist Expose

Photoresist Development

Oxide Etch

Photoresist Strip

Comment

The process described here is an N well process since it only has an N-well. P well and Twin Tub process also exist.

N-well

Active Region

Polysilicon

P implant

N implant

Contact window

Metal

Inter-digitated power and ground rails can be routed using only one metal layer

ASIC designer:

⇒ placement:

Cells are placed in one or several equal length lines with inter-digited power and ground rails.

⇒ Routing

In the routing channels between the cells we route metal1 horizontally and metal 2 vertically.

Layout Abstract Routing

#1 Metal Layer

ABSTRACT ROUTING LAYOUT

Circuit Diagrams

Circuit Diagram

Kanmas
Circuit schematic diagram
Schematic diagram
Circuit schematic
Schematic

Gat level circuit diagram

Transistor level 设备

= A circuit diagram where all the elements are transistors.

Stick Diagram Vs Mask level Layout Vs Cross section diagram

Adder

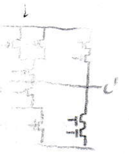
[Binary Adder]

$S = A \oplus B \oplus C_{in}$

$C = AB + AC_{in} + BC_{in}$

$\rightarrow = (A \oplus B) C_{in} + AB$

Can share the term $A \oplus B$



[Look from a different angle]

Generate signal: $G = A.B$

Propagate function: $P = A + B$

We use $P = A \oplus B$ since it shows the same behaviour and also can be used to generate the sum term

$S = P \oplus C_{in}$

$C = G + P.C_{in}$

[Multi-bit adder]

Ripple carry / carry propagate

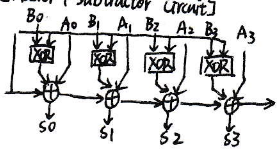
T_d (worst) $= (N-1) T_{carry} + T_{sum}$

[Carry chain Optimisation]

$S = ABC_{in} + (A + B + C_{in}) \cdot C'$

↳ For each of the full adders the c signal is coming earlier than the sum signal.

[Adder / Subtractor Circuit]



Carry-Lookahead Adder (CLA)

To predict C_k earlier than $T_c \times k$

A	B	Cin	C	
0	0	X	0	Kill
0	1	C	C	Propagate
1	0	C	C	Propagate
1	1	X	1	Generate

$P = A + B$ or $(A \oplus B)$

$g = A.B$

$\rightarrow C_4 = 1$

↳ g_3

$g_2 P_3$

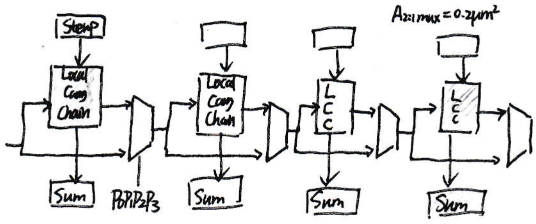
$g_1 P_2 P_3$

$g_0 P_1 P_2 P_3$

$C_{in} P_0 P_1 P_2 P_3$

$C_4 = g_3 + g_2 P_3 + g_1 P_2 P_3 + g_0 P_1 P_2 P_3 + C_{in} P_0 P_1 P_2 P_3$

Carry-Bypass Adder (CBA)



$A_{xor} = 1\mu m^2$

$A_{AND} = 0.5\mu m^2$

$A_{FA} = 2\mu m^2$

$A_{2:1 mux} = 0.2\mu m^2$

$A_{setup} + A_{LCC} + A_{mux} = 14.2\mu m^2$

↳ $4(A_{and} + A_{xor}) = 4 \times 1.5 = 6\mu m^2$

$A_{LCC} = 4 \times 2 = 8\mu m^2$

$A_{mux} = 4 \times 0.2 = 0.8\mu m^2$

$A_{all} = 3 \times 14.2 + 14 =$

$t_{and2} = 1$

$t_{xor} = 2$

$t_c/bit = 4$

$t_{sum} = 2$

$t_{2:1 mux} = 1$

$t_{setup} = 2$

$t_{FA} = 4 \times 4 = 16$

$t_{select} = P_0 P_1 P_2 P_3 = 2$

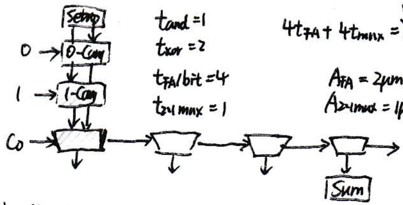
$t_{mux} = 3 \times 1 = 3$

$t_{all} = 2 + 16 + 3 + 12 + 2$

(setup, t_{FA}, t_{mux}, $3 \cdot t_c$, t_{sum})

Delay $= t_{setup} + \max\{t_{select}, 4 \times t_{FA}\} + 3 \times t_{mux} + 4 \times t_{FA} + T_{sum}$

Linear carry-select adder: structure



$t_{and} = 1$

$t_{xor} = 2$

$t_{FA}/bit = 4$

$t_{2:1 mux} = 1$

may be no setup by using ripple adder

$4t_{FA} + 4t_{mux} = 2 + 16 + 4 = 22$

$A_{FA} = 2\mu m^2$

$A_{2:1 mux} = 1\mu m^2$ ⇒ $8A_{FA} + 5A_{mux} = 21$

{4 for Sum, 1 for Carry}

Booth's algorithm

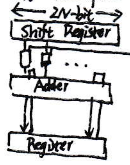
r_i	r_{i-1}	$r_{i-1} - r_i$	
0	0	0	Add Zero
1	0	-1	Subtract
0	1	1	Add
1	1	0	Add Zero

```
    0110     6
   x1001    -5
 11111010   -1
 0000000     0
 000110      1
 11010      -1
 11100010
```

Modified Booth Algorithm

r_{i+1}, r_i	r_{i-1}	$r_{i-1} + r_i - 2r_{i+1}$	
0 0	0	0	Do nothing
0 0	1	1	Add 1*
0 1	0	1	Add 1*
0 1	1	2	Add 2*
1 0	0	-2	Sub 2*
1 0	1	-1	Sub 1*
1 1	0	-1	Sub 1*
1 1	1	0	Do nothing

Shift Multiplier

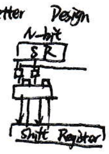


One bit of the multiplier

{2N-bit adder, 2N-bit register}

Shift the multiplicand to the right

Better Design

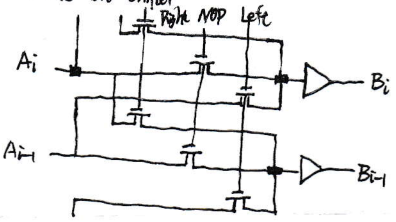


{N-bit and gate, N-bit adder}

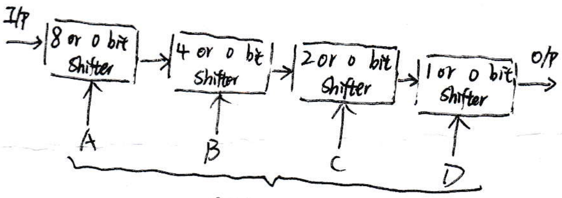
Shift the result to the right

Don't need to shift the multiplicand and the multiplier so that we save the bit of register and the adder

One bit shifter



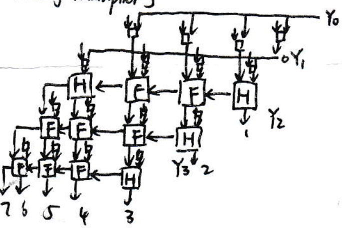
Programmable shifter



Control Lines

Multiplier

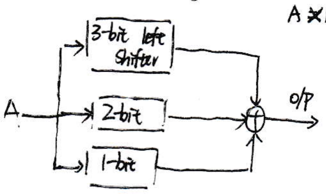
[Array Multiplier]

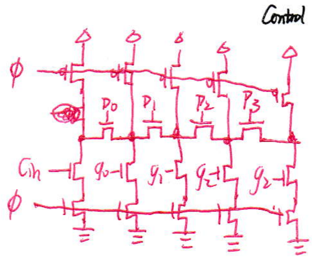


N bit ⇒ {N(N-2) FAs, N HAs, N^2 ANDs} $\approx \frac{N}{2}$ FA

Multiplication using Shifter

$A \times 14 = A(2^3 + 2^2 + 2^1)$



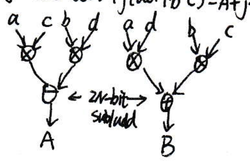


Precharge internal nodes to avoid charge-sharing problem

Complex numbers and addition

$\begin{cases} x = a + j \cdot b \\ y = c + j \cdot d \end{cases}$

$xy = (ac - bd) + j(ad + bc) = A + jB$



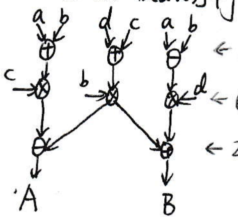
Complexity: $4[N(N-2) FA + \frac{N}{2} FA + N^2 AND]$ $+ 2$ (2N-bit) FA

[Reduce the complexity]

$xy = (ac - bd) + j(ad + bc)$

$= [ac - bd + bc - bc] + j[d(a-b) + b(c+d)]$

$= [c(a+b) - b(d+c)] + j[d(a-b) + b(c+d)]$



← N bit Adder

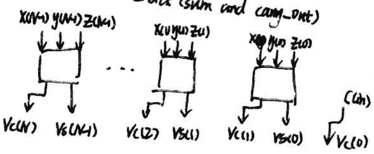
← (N+1) bit multiplier

← 2(N+1) bit adder

Redundant Arithmetic

The basic idea is to perform an addition of three vectors (or inputs) using an array of 1-bit adder - BUT without propagating the carries.

→ A full adder takes 3 inputs (a, b, carry-in) and converts them to 2 bits (sum and carry-out)




```
X    0 1 1 1 0 1 0 0
Y    0 0 1 1 1 0 1 1
Z    1 0 1 0 1 0 1 0
----------------------
Vs    1 1 1 0 0 1 0 1
Vc  0 1 1 1 1 0 1 0 0
```

(Cout) Carry is not added but saved in Vc (Cin)

E.g. X=1010 y=0111 z=1000

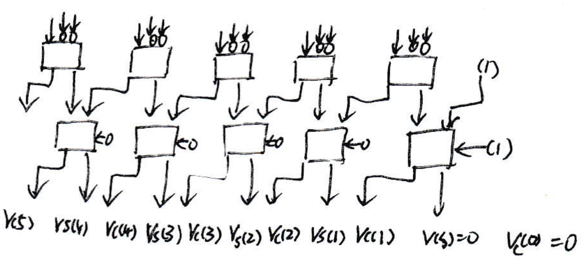
```
   1010
  +0111
   1000
Vs= 00101
Vc= 10100
```

For conventional addition: 4T+5T=9T

Vs+Vc = 00101 + 10100 = 11001

Timer required ⇒ T+5T = 6T

(MMP)

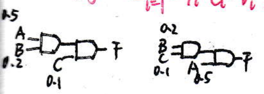


Low Power Design

$P_d = C_L V_{DD}^2 \cdot f$

Internal nodes are making power consuming transitions but output node is not showing the effect

$P_{avg} = \left(\sum_{i=1}^{\#\text{ of node}} \alpha_{Ti} \cdot C_i \cdot V_i\right) \cdot V_{DD} \cdot f_{clk}$ (In general $V_i = V_{DD}$)



The higher probability signals should be introduced at a later stage in the logic pipeline to reduce toggling Activity!

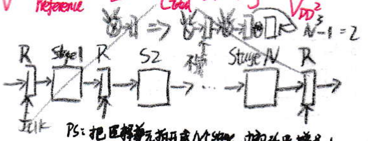
Low-Power Design Using Pipelining Approach

$P_{reference} = C_{total} \cdot V_{DD}^2 \cdot f_{clk}$

$P_{pipelining} = [C_{total} + (N-1) \cdot C_{reg}] \cdot V_{DD,new}^2 \cdot f_{clk}$

☆ $\frac{P_{pipeline}}{P_{reference}} = \left[1 + \frac{C_{reg}}{C_{total}}(N-1)\right] \frac{V_{DD,new}^2}{V_{DD}^2}$

$N-1 = 2$



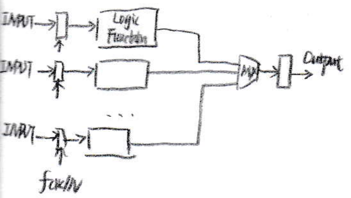
PS: 把逻辑单元拆开成N个stage，内部延迟增加！

Logic function F(input) has to be partitioned into N successive stage each having approximately equal delay.

A total of (N-1) register arrays have to be introduced in addition to the original input and output registers

All registers are clocked at f_{clk} (though it is possible to increase the clock speed to $N \cdot f_{clk}$ and thus, delay for each logic block can be increased by a factor of N without sacrificing throughput 吞吐量)

Parallelism



Assume that the consecutive input data arrive at the same rate as in the single-stage case.

The input data are routed to all registers in N processing blocks.

- Gated clock signals with clock period of NT_{clk} are used to load each register every N_{clk}
  - The individual clock signals are skewed by T_{clk}, such that each one of the N consecutive data is loaded into a different input register.
  - Each register is thus clocked at a lower frequency f_{clk}/N, the time allowed to compute the function for each input data is increased by a factor of N

The supply voltage can be reduced until the path delay equals the new clock period of NT_{clk}

$P_{parallel} = \left[1 + \frac{C_{reg}}{C_{total}}\right] \cdot C_{total} \cdot V_{DD,new}^2 \cdot f_{clk}$

$\frac{P_{parallel}}{P_{reference}} = \frac{V_{DD,new}^2}{V_{DD}^2}\left[1 + \frac{C_{reg}}{C_{total}}\right]$

Pipelining vs Parallelism

Pipelining:
- Increase Area
- Increase of latency from 1 cycle to N cycles

Parallelism:
- Increase area and latency
- Silicon area will grow faster than the number of processor because of signal routing and the overhead circuitry.

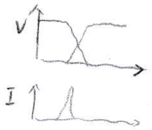
Short Circuit Power Dissipation

Power dissipation due to non-ideal switching characteristic of CMOS logic gates

☆ The input of a real life logic gate (e.g. an inverter) is actually driven by input voltage waveforms that has finite rise and fall time.

Both the NMOS and CMOS transistors may conduct simultaneously for a very short amount of time during switching, forming a direct current path between the supply and the ground.

↳ May be reduced by making the output voltage transition time longer and/or making the input voltage transition time smaller. should be balanced carefully.



Supply current to charge up the output capacitance

Leakage Power Dissipation

NMOS and PMOS transistors generally have nonzero reverse leakage and subthreshold currents

So even when the gate is idle, it still draws a very small amount of current

- For a chip having millions of transistors, this leakage power dissipation may contribute significantly.

Two main components: biggest concern in deep submicron

{ Reverse gate leakage current
  Subthreshold current

☆ Two main components of Leakage Power

① Reverse gate leakage current
- Tunnelling of electrons through gate oxide.

② Subthreshold current

A weak current that flows due to carrier diffusion between the source and drain of a transistor in weak inversion region (V_{GS} is less than the threshold voltage but not exactly zero)

If V_{GS} is smaller than but very close to the threshold voltage then the subthreshold power dissipation may become comparable in magnitude to the switching power dissipation

[Leakage Reduction]

Process level
- Channel engineering
- Well engineering

Circuit level: Control voltages at different device terminals
- Standby leakage control using transistor stack
- Multiple threshold design
- sleep transistor insertion

Why Low Power Design?

① Increasing prominence of portable and mobile system
- Requirement of low power dissipation to maximize battery life

② High-performance microprocessors and DSPs require very complex chips with high clock frequency.
- Power dissipation and hence temperature increase linearly with frequency and complexity

③ Reliability
- Close correlation exists between the peak power dissipation of digital circuits and reliability.

④ General-purpose microprocessors and DSPs are not a good solution.
- Each instruction execution typically requires a number of clock cycles and thus, the power dissipation is at least 1 or 2 orders of magnitude higher than the hardware approach.

Timing Constraint : The data applied to the D-flip-flop should be stable for at least Tsetup and should stay stable for at least Thold.

Setup time (T_{setup}) : is the minimum amount of time the data should be stable before the application of the clock signal.

Hold time (T_{hold}) : is the minimum amount of time the data should be stable after the application of the clock signal.

[Setup Timing Constraint]

$$T_{clk-min} \geq T_D + T_{IC-max} + T_{setup}$$

[Holding Timing Constraint]

$$T_{hold} \leq T_D + T_{IC-min}$$

The role of interconnect

Buses are implemented as long metal lines on a silicon wafer transmitting data using electromagnic waves (finite speed limit)

In addition to data transfer Metal lines (interconnect) are also used power distribution networks and for clock distribution network in synchronous design.

Interconnect networks are have became increasingly complicated with greater integration

The Impact of Interconnection in Modern Technologies

① Limit Performance

Signals cannot travel across the entire die within a global clock cycle.

For example 6-10 clock cycles are needed to transfer data on a bus at 50nm technology

Unpredictability in signal propagation time has serious consequences for performance and correct functioning of synchronous digital circuits

② Consume up to 50% of chip power

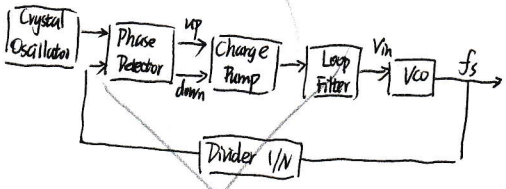
③ Affect system reliability (e.g. crosstalk & errors)

$$D = 0.4 * R * C$$

Phase-Locked Loop

PLL is typically used to generate the high frequency required by SoC. A PLL takes an external low-frequency reference crystal frequency signal and multiples its frequency by a rational number I/N

The reference clock is typically generated off-chip from an accurate crystal reference. The reference clock is compared to a divided version of the system clock (i.e. the divided clock) using a phase detector that compares the phase difference between the signals and produces an Up or Down signal when the local clock ge lags or leads the reference signal. It detects which of the two inputs signals arrives earlier and produces an appropriate output signal.



Phase detector : Determine the relative phase and frequency difference between two incoming signals and output a signal that is proportional to this difference.

Charge Pump : It converts the Up/Down pules into an Analog voltage that controls the VCO.

Loop Filter : Low-pass filter that removes the high-frequency components from the Vco control voltage and smoothens out its response, which results in a reduction of the jitter

Clock Skew

Static point-to-point variation (of clock arrival time to FFs)

① Designed variations - mismatch in buffer load sizes, interconnect lengths

② Process variations - process speed across die yielding different Leff T_{ox}

③ Temperature gradients

④ IR voltage drop in power supply

Clock Jitter

is the variation in clock edge timing between clock cycle. (cycle to cycle variation)

① Variation in PLL oscillation frequency (PLL is highly sensitive to intrinsic device)

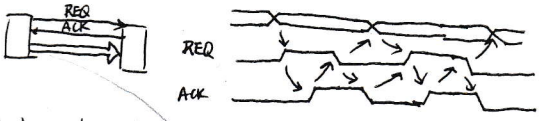
② High Frequency Power supply variations (Caused by instantaneous IR drops along the power grid)

③ Variation in capacitive load (Due to coupling between the clock lines and adjacent signal wires)

$$T_{clk-min} \geq T_D + T_{IC-max} + T_{setup} + 2 \cdot Jitter - Skew$$

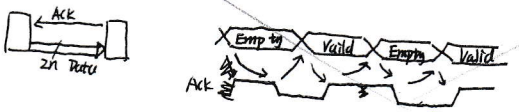
$$T_{hold} \leq T_D + ~~T_{IC-max}~~ T_{IC-min} - 2 \cdot Jitter - Skew$$

~~Bound~~ Bundled data Protocol (single rail)



data should be valid preceding REQ

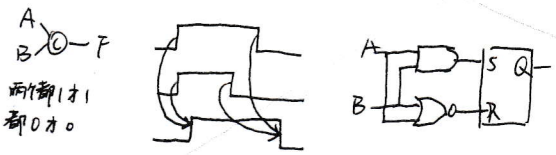
4-phase dual rail protocol



Delay Insensitive

Each change is acknowledged

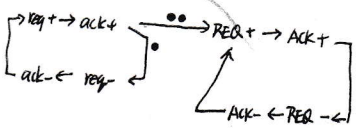
Problems : Glitches, hazards



若A和B为1，则F为1

若A和B为0，则F为0

If left cycles goes fast and right cycle lags, then arc ack+ → REQ+ accumulates tokens



Boundedness

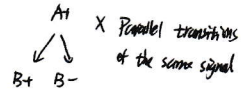
Caused by one-sided dependency

STG is bounded (safe) if no place or arc can ever contain more than one token

Consistent

A+ ⋯→ A+ Two subsequent up-transitions without a down-transition in between

A- ⋯→ A- Two subsequent down-transitions without an up-transition in between



X Parallel transitions of the same signal

Boundedness + Consistent State Assignment ⇒ Finite Binary State Graph

State Coding 补

{ Concurrency reduction
  Signal insertion

Output Persistency

Output disables output (metastability or glitch)

Metastability is a condition where the voltage level of a signal is at an intermediate level (neither 0 or 1) and which may persist for an indeterminate amount of time

$$MR = W \times F_c \times F_D^2$$

$$MTBF = \frac{e^{\frac{S}{\tau}}}{W \times F_c \times F_D}$$

1FF, No $e^{\frac{S}{\tau}}$

2FF, $S = F_c$

3FF, $S = 2F_c$