

& concatenation
msb & data[n-1:1]
output

note pad
fichier .vhdl

data[n-2:0] & data[0]
n-1

dernier élément avant parenthèse ; pas de point virgule
ligne

Data <= (others => '0') chaque bit de data

$$A \overset{\text{XOR}}{\oplus} B \overset{\text{XOR}}{\oplus} \text{Cin} = S$$

$$(A \text{ and } B) \text{ OR } ((A \text{ XOR } B) \text{ and } \text{Cin}) = \text{Cout}$$

entrée: A B Cin

sortie: S Cout

suivre le schéma RTL clair

ordre de priorité comme math or et xor ?

synoptique pour représenter il peut y avoir
différent niveau d'abstraction.

architecture nager très clair et très précise

et mettre à jour le schéma RTL avec modifications
de code

contrainte de timing au placement
horloge

espacer deux signaux
qui sont verticalement
perturbés

simulation suture de test bench

Vérifier que la synthèse a bien fait ce que l'on voulait avec le rapport de synthèse

Placer les sondes avec l'ILA

placement routage sur implémentation

pour toutes les possibilités d'entrée quelle sont les sorties simulation

process sans liste de sensibilité exécuté dès le départ et exécuté indéfiniment.

UUT Unit Under Test

06 51 05 58 06

07 62 26 38 25

HDL Codes matlab

Siemens catapult

Migen avec python

ingénieur système

outil HLS pas étudié ici

système vérilog pour test bench

retro ingéniering du bitstream sécurité du bitstream

pas trop grand public mais militaire OUI

gogs git hub

FPGA Arvans

car exécution ne dépend pas d'accès mémoire externe

Baïte routage, que des fils, mais passent par des Baïte de routage
comme des portes logiques

chemin le plus long = chemin critique
on cherche à le minimiser d'où l'effort de réduction
d'équations logiques

15 ps

25 ps C1

20 ps S1

schéma de data flow flux de données

synthèse code VHDL en schéma logique

Slice entrée CENTER INTER \Rightarrow Baïte de
routage

casi chain pratique pour réaliser des compteurs
spécifiques à cette carte là

Une slice un logical élément
intel CLB

latence Microsemi autre nom regroupement de composants
de logique de bas niveau

multiplicateur mini baïte de routage

opération comptage temporisation
porte en dur bonne performance en latence

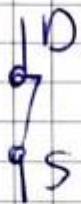
fpga 4 fms

Logic Array Block

MOSN

$V_{gs} >$

$V_{gs} > V_{th}$



$V_{gs} < V_{th}$



PMOS

$V_{gs} < -V_{th}$

substrat semi conducteur silicium

bombardé avec électrons \rightarrow photons laser on oxydant au dessus la surface

ctrl R notation comparant

latence circuit

comportement durant la porte logique

600 85,0

300 200 600 60ms

changement d'état des transistors ils mettent certain temps
il y a un court instant où les deux transistors sont fermés

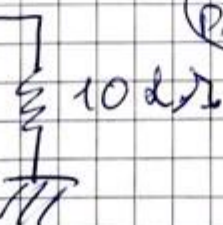
Power Budget

transistor p un peu plus lent

$V_{thp} = -0,5V$

$V_{thn} = 1,1V$

source



Pmax 11

si $V_d = 0$ $V_{gs} = 0 \rightarrow$ arrêt

diode 120

pull down à zero

LUT Look up Table table de vérité
possibilité du comportement d'une mini
nano unité de ^{mémoire} ~~analog~~

A B C D	logical ^{output} input	Address	Content RAM
0000	0	0000	0
0001	0	0001	0

ASIC Application Specific Integrated circuit
fondé sur fpga synapsys cadence

antel passage de VHDL asic

c X B fpga

X

500 600 millions euros

1 millions d'essais

chiffage à 5 millions

20 30 millions euros

Millions sur asic

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$$S = A \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

$$\text{Donc } S = \bar{C} (A \cdot B + A \cdot \bar{B}) + C (A \cdot B + \bar{A} \cdot B)$$

$$S = C \cdot B + \bar{C} \cdot A$$

A B		0 0		0 1		1 1		1 0	
C	0	0	0	0	0	1	1	1	1
	1	1	1	1	1	1	1	1	1

$$S = \cancel{A} B + C \cancel{B} + \bar{C} A$$

et pas 0

mémoire SRAM ou FLASH → populaire
 fpga à base de fusible
 encore plus compliqué de faire
 de la vraie ingénierie

apparent avec très très peu d'unité
 difficile à caractériser résistance radiation.

Sarles Silbers
 3D + exco-trail

Fpga base ^{mémoire} volatile ou ^{mémoire} non volatile

SRAM → Meilleure densité, accès plus rapide
 eMMC pour carte SD que Flash

Full subtractor

$$0xFF = 11111111$$

$$-128 + 69$$

A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

more carry

$$-2 + 1$$

AB Bin	00	01	11	10	D
0	0	1	0	1	
1	1	0	1	0	

Bin $\bar{A} \bar{B}$ XOR Bin $\bar{A} B$ XOR Bin $A \bar{B}$ XOR Bin $A B$

AB Bin	00	01	11	10	Bout
0	0	1	0	0	
1	1	1	1	0	

$$\text{Bin } \bar{A} \bar{B} \text{ XOR } \bar{A} B \text{ Bin} + \text{Bin } B$$

$$\text{Bin } \bar{A} \bar{B}$$

$$A B \text{ Bin } \bar{A} \bar{B}$$

$$\text{Bin } \bar{A} \bar{B} \text{ XOR Bin } \bar{A} B \text{ XOR } A B \text{ Bin XOR } A \bar{B} \text{ Bin}$$

Nithade 2

$$D = \bar{A} \bar{B} \text{ Bin} + \bar{A} B \bar{\text{Bin}} + A \bar{B} \bar{\text{Bin}} + A B \text{ Bin}$$

$$= \text{Bin} (\bar{A} \bar{B} + A B) + \bar{\text{Bin}} (\bar{A} B + A \bar{B})$$

A	B	\bar{A}	\bar{B}	$\bar{A}\bar{B}$	AB	$\bar{A}\bar{B} + AB$	$\bar{A}B$	$A\bar{B}$	$\bar{A}B + A\bar{B}$	$\bar{A}\bar{B} + AB$
0	0	1	1	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1	1	0
1	1	0	0	0	1	1	1	1	1	1

$$D = \text{Bin}(A \text{ XNOR } B) + \bar{\text{Bin}}(A \text{ XOR } B)$$

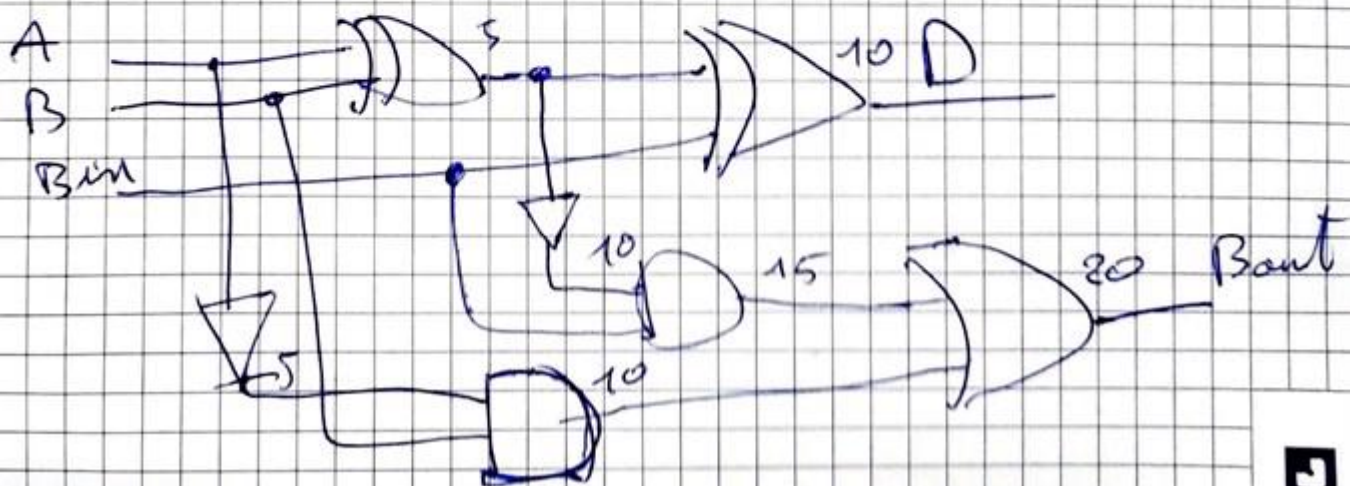
$$D = \text{Bin XOR } (A \text{ XOR } B)$$

$$\text{Bout} = \bar{A} \bar{B} \text{ Bin} + \bar{A} B \bar{\text{Bin}} + \bar{A} B \text{ Bin} + A B \bar{\text{Bin}}$$

$$\text{Bin} (\bar{A} \bar{B} + \bar{A} B + A B) + \bar{\text{Bin}} \bar{A} B$$

$$= \bar{A} B (\bar{\text{Bin}} + \text{Bin}) + \text{Bin} (\bar{A} \bar{B} + A B)$$

$$\bar{A} B + \text{Bin}(A \text{ XNOR } B)$$



logique numérique synchrone
notions de temps dans le calcul
mécaniser au minimum ce que l'on calcule
résultat pas disponible à $t=0$ (zéro)

Efinix
tiny fpga z80
TP le plus important

bascule
registre

compteur registres à décalage

Machines à état (Finite State Machine)

latch conserve la valeur

S	R	Q	\bar{Q}
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

latch la $(S=1, R=0) Q=1$ vaut 1

$(S=0, R=0)$ latch $Q=1$ reste à 1

$(S=0, R=1) Q=1$ vaut 0

$(S=0, R=0)$ latch $Q=1$ reste à 0

cas illégal pas utilisation
de la bascule

Dépend de l'antécédent

contre réaction se lance mise en saturation
on veut déclencher la mise en saturation au moment
où on le souhaite.

le registre

set synchrone avec ck

reset asynchrone

c'est au front montant



1st reset nat. de temps
main enclenche perturbation