ALU of GPS Processor

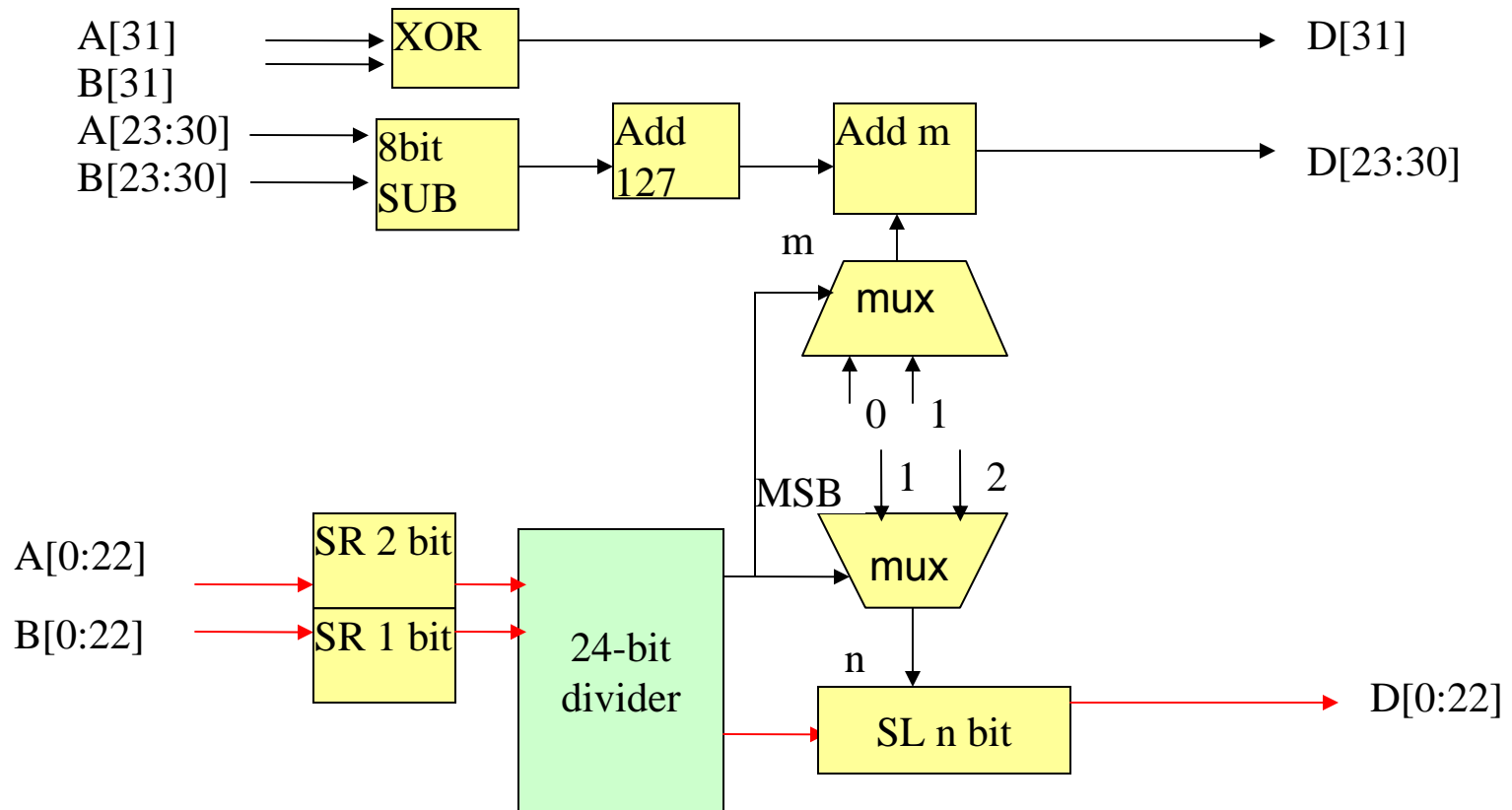## Cheuk Chon & Zhibin Wu

ALU Overview

- IEEE 754 format
- Floating point operations
  - Add/sub
  - Multiply
  - Divide
- Integer Arithmetic Unit
  - Add/sub
  - Multiply
  - Divide
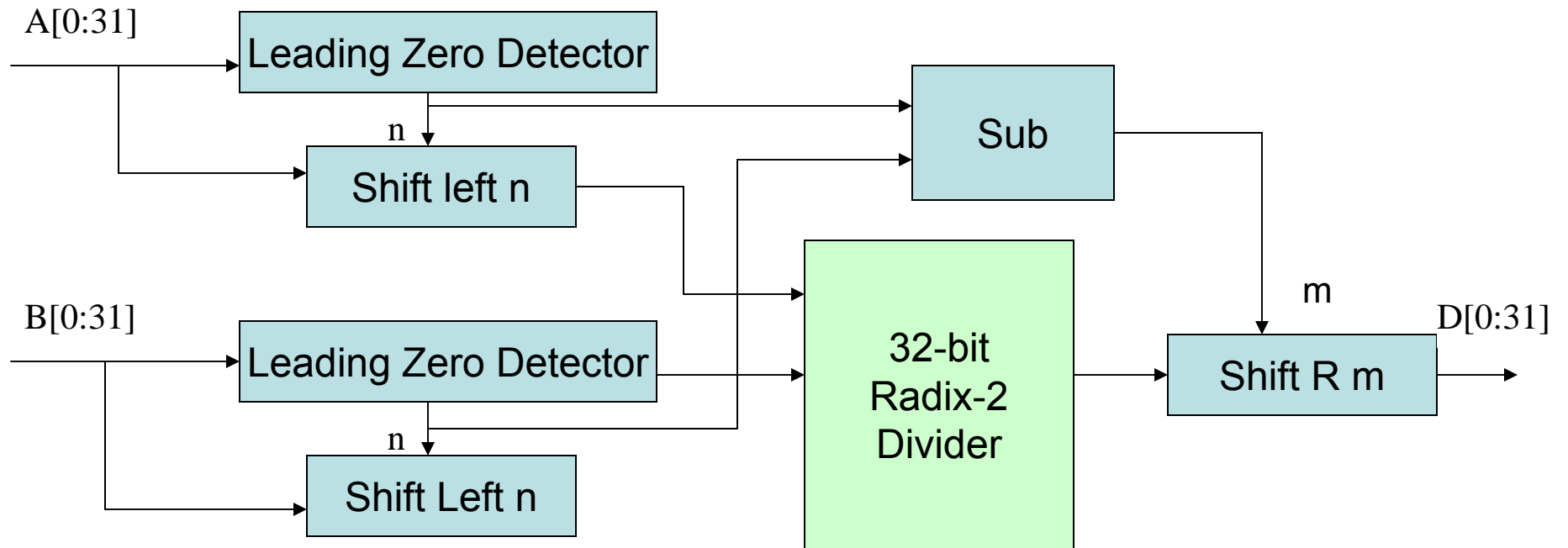 - Shift(Arithmetic and logical)
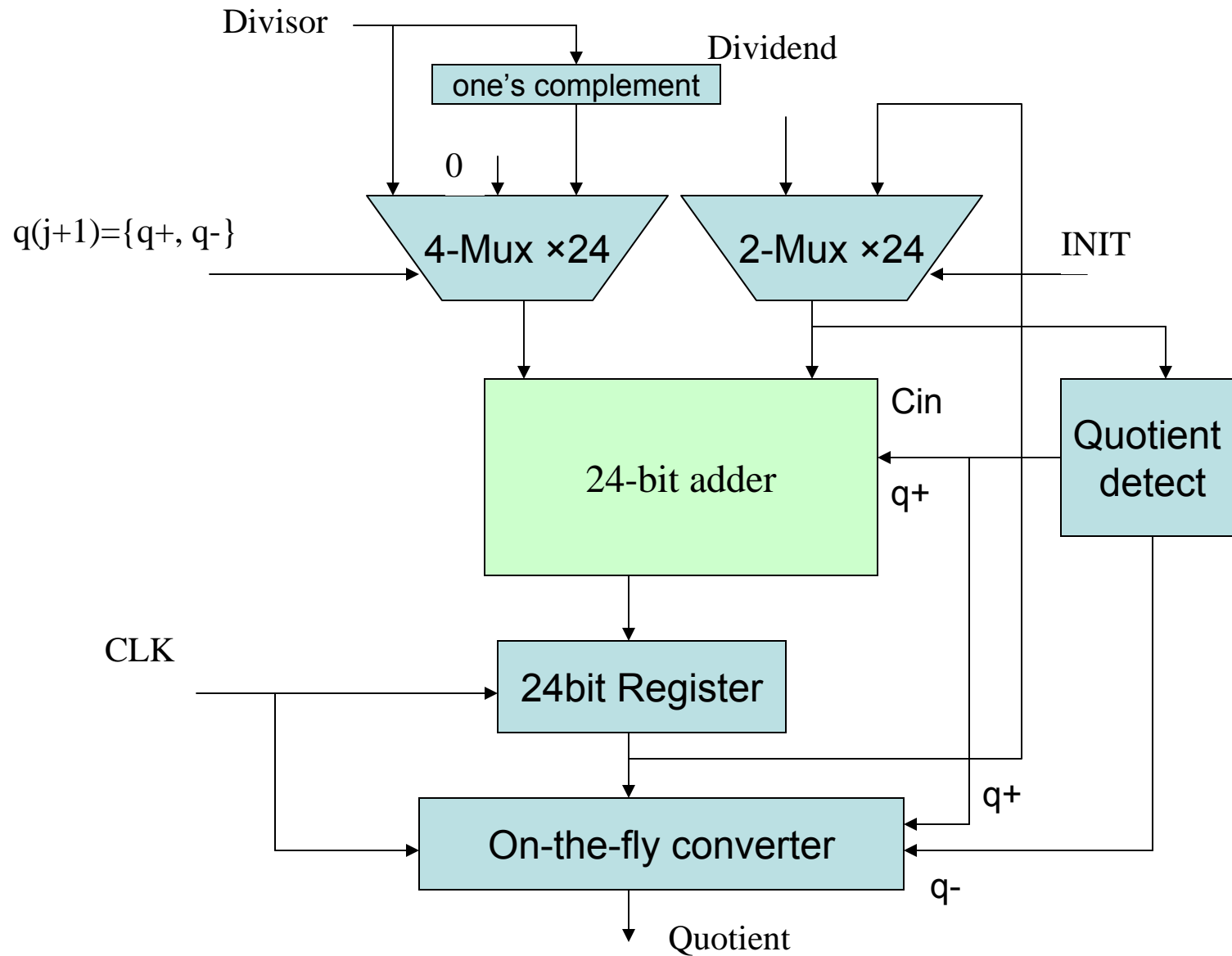
Divider in ALU

Zhibin Wu

Top-level diagram(floating point)

Integer Divider

A[0:31]

Leading Zero Detector

n

Shift left n

Sub

B[0:31]

Leading Zero Detector

n

Shift Left n

32-bit Radix-2 Divider

m

Shift R m

D[0:31]

Radix-2 Divider

Divisor

Dividend

one's complement

0

$q(j+1)=\{q+, q-\}$

4-Mux ×24

2-Mux ×24

INIT

Cin

24-bit adder

q+

Quotient detect

CLK

24bit Register

q+

On-the-fly converter

q-

Quotient

Chip Area

- Floating point divider
  - 0.225 mm × 0.225 mm
- Integer Divider
  - 0.27 mm × 0.27 mm
- 32 bit barrel shifter
  - 0.27 mm × 0.27 mm
- Divider use small area because a recursive algorithm is used
  - Trade off: area *.vs.* speed

Basic cell layout

- For reference
  - A one-bit full adder: 200λ * 100λ
  - 1-bit Half adder :100 λ * 100 λ
  - Basic area for register: 200 λ * 50 λ
  - Basic area for 2-to-1 mutiplexer: 50 λ * 50 λ

Power Consumption Estimation

- From specification:
- One 8-bit full adder cost 0.9968mw
  - 24 bit full adder: 2.9904mW
  - 32 bit full adder: 3.9872mW
- Consider other components and duty 20%:
  - Floating point adder: <10mW
  - Integer divider: <8mW
- GPS project's specification has no low-power requirement.
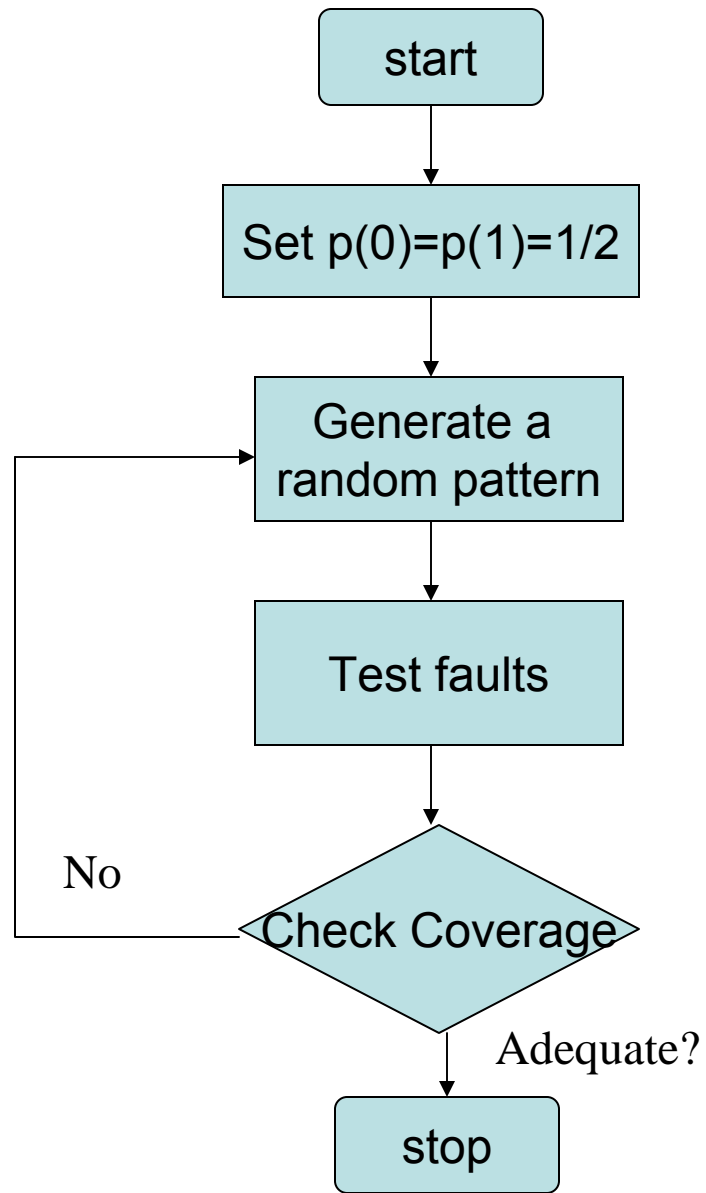
Timing Analysis for Full Adder

- Carry-ripple adder delay

  - Basic on results for 8-bit carry ripple: 3.80ns

  - 24-bit carry ripple: 11.4ns

  - 32-bit carry ripple: 15.2ns

- CPU Clock: 10MHz, 100ns

- Full adder can perform the operation within one clock cycle.

Critical Path & delay

- For floating point divider
  - Mantissa part is not as slow as fraction part.
  - 24 clock cycles
  - Delay is 2400ns
- For integer divider
  - 33 clock cycles
  - Delay is 3300ns
- Delay is depend on the number of clock cycles, other delays can be neglected.

Testing strategy for ALU

- 64 inputs,impossible to exhaust all input patterns
- Strategy:
  - Random automatic test pattern generation
  - Specify some pattern to cover "exception" faults
- Adding Testability
  - Adding control pins to set internal state
  - Adding observability

```
            ┌──────────┐
            │  start   │
            └────┬─────┘
                 │
                 ▼
       ┌────────────────────┐
       │ Set p(0)=p(1)=1/2   │
       └─────────┬──────────┘
                 │
                 ▼
       ┌────────────────────┐
  ┌───▶│    Generate a       │
  │    │   random pattern    │
  │    └─────────┬──────────┘
  │              │
  │              ▼
  │    ┌────────────────────┐
  │    │    Test faults      │
  │    └─────────┬──────────┘
  │              │
  │              ▼
  │          ╱────────╲
No│         ╱  Check    ╲
  └────────⟨  Coverage   ⟩
            ╲           ╱
             ╲─────────╱
                 │ Adequate?
                 ▼
            ┌──────────┐
            │   stop    │
            └──────────┘
```

GPS

32 bits Floating Pointing Multiplier/ Adder/Subtractor
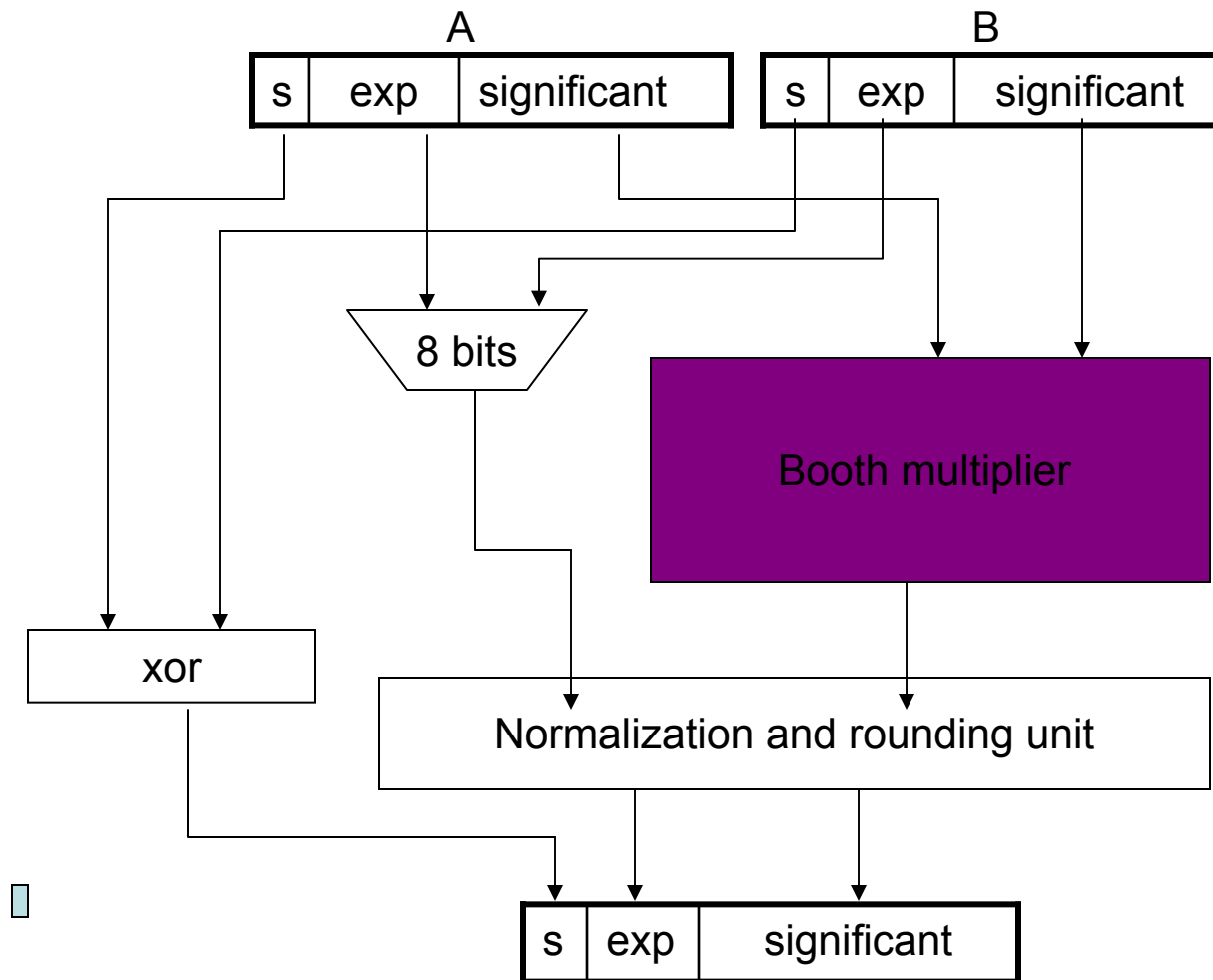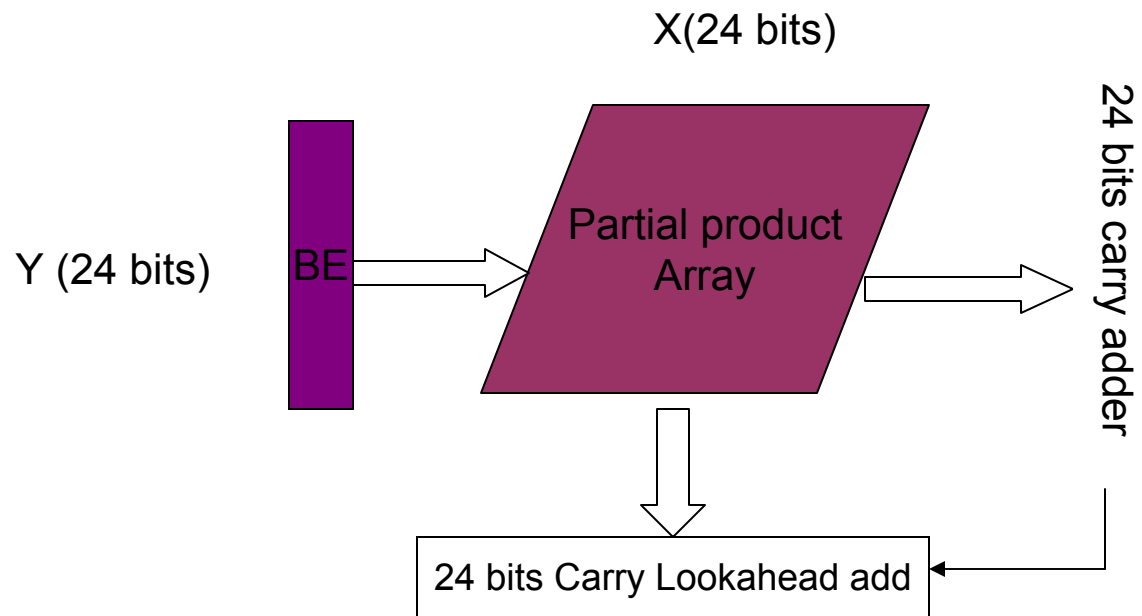
Cheuk Hon Chan
12/20/02

FPU Introduction

- Using IEEE 754 floating point format (single precision)

- Using booth algorithm for multiplier.

- Only need 12 rows of partial products to generate the result

- The area for multiplier is 0.7 x 0.7 mm ^ 2.

- The area for adder / subtractor is 0.4 / 0.4 mm ^ 2.

- The unit is very fast.

32 Bits Floating Point Multiplier

A

s	exp	significant

B

s	exp	significant

8 bits

Booth multiplier

xor

Normalization and rounding unit

s	exp	significant

Floating Point Multiplier

X(24 bits)

Y (24 bits)

BE

Partial product Array

24 bits carry adder

24 bits Carry Lookahead add

Booth Multiplier – Booth encoder

Yi +1	Yi	Yi-1	Encoded digits	Explanation
0	0	0	0	Add 0
0	0	1	+1	Add x
0	1	0	+1	Add x
0	1	1	+2	Shift left x by one digit and add
1	0	0	-2	Add 2's complement of x and shift left by one digit
1	0	1	-1	Add 2's complement of x
1	1	0	-1	Add 2's complement of x
1	1	1	0	Add 0

8 bits Booth Multiplier

Power Consumption

The power for a 8 bits carry-lookahead adder is 0.996 mW.

A 24 bits carry-lookahead adder is 3.9 mW.

One full adder in booth multiplier block is 88 uW.

One half adder in booth multiplier block is 20.56uW

Each mux block is 9.7 uW.

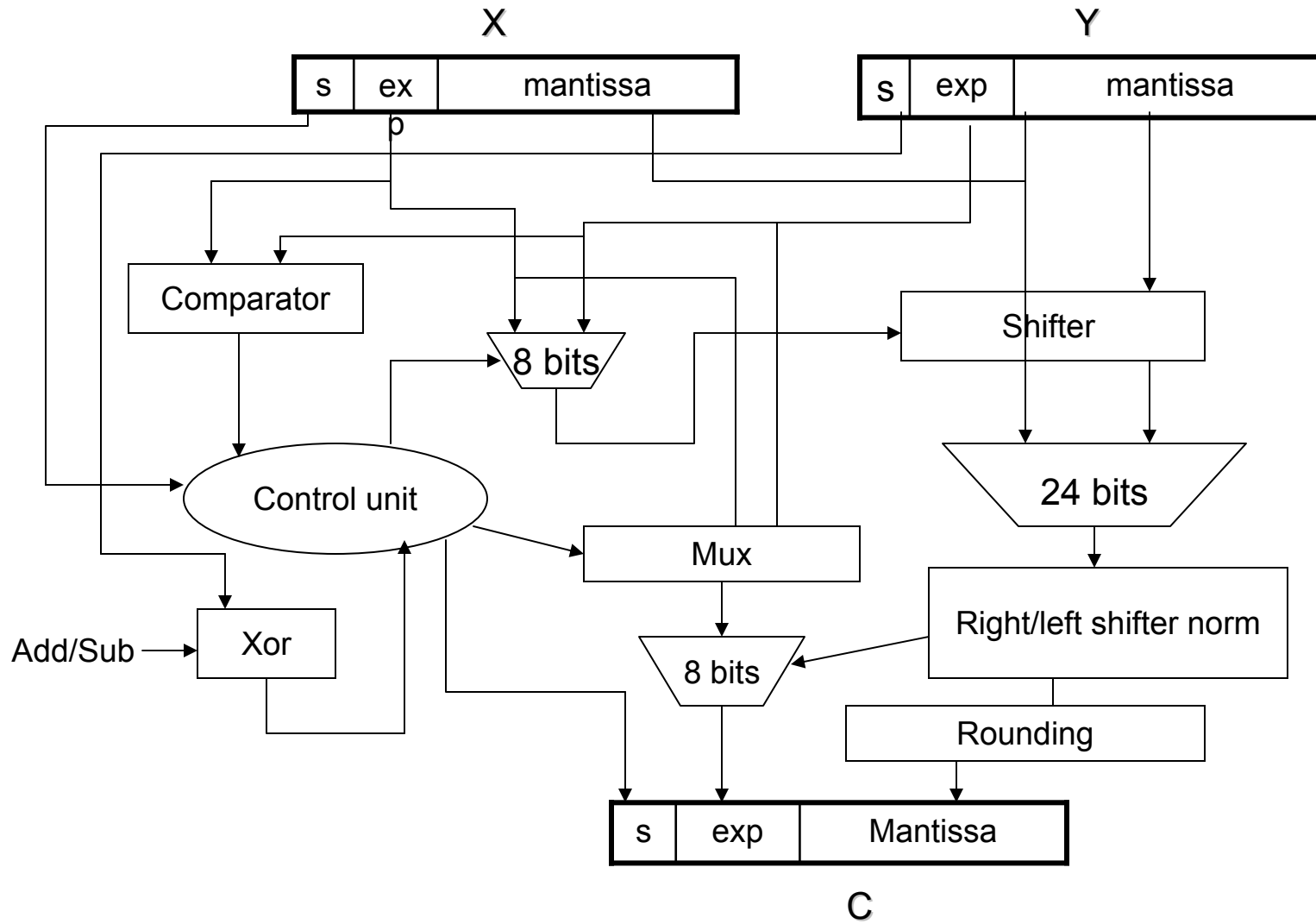Each booth encoder (BE) block requires 0.4 mW.

The whole booth multiplier block is 7.8mW.

The floating point multiplier consumes 11.3 mW

Timing analysis

- A 8 bits carry lookahead adder is 1.3 ns.
- A 24 bits carry lookahead adder is 4 ns.
- Each full/half adder block in booth multiplier is 0.7 ns.
- Each mux block in booth multiplier is 0.35 ns.
- Each booth encoder unit is 0.29 ns.
- The booth multiplier block is 12.05 ns.
- The floating point multiplier is 16.25 ns.

32 bits Floating Adder / Subtractor

X

s	ex p	mantissa

Y

S	exp	mantissa

Comparator

8 bits

Shifter

Control unit

24 bits

Mux

Right/left shifter norm

Add/Sub → Xor

8 bits

Rounding

C

s	exp	Mantissa

Power Consumption / Timing analysis

- A 8 bits comparator needs 0.650 mW.

- Then, a 24 bits carry-lookahead adder is 3.9 mW.

- The floating point adder/subtractor is 9.65 mW.

- The comparator is 1.24 ns.

- A 24 bits carry lookahead adder is 4 ns.

- The whole floating point adder/subtractor is 9.65 ns.

Conclusion

- The overall area for ALU is 2mm x 2mm.
- The total power consumption < 40 mW
- The ADD SUB, SHIFT, MULTIPLY instruction is done within one clock cycle.
- The DIV is using the multiple clock cycles to perform the operation
- The main test method for ALU uses ATPG.