Project2 Approximate Adder Design

Digital Integrated Circuit Design

TA1: Yuxuan Qin - qinyuxuan@sjtu.edu.cn;
TA2: Zihan Lian - lianzihan@sjtu.edu.cn;

Outline

- **实验概述**
  - **加法器简介**
  - 近似计算加法器
- 实验要求
- 附录

半加器

- 两输入a、b，两输出s、cout

$$c_{out} = a \cdot b$$
$$s = a \oplus b$$

a	b	c_{out}	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

全加器

- 两输入a、b，进位输入cin

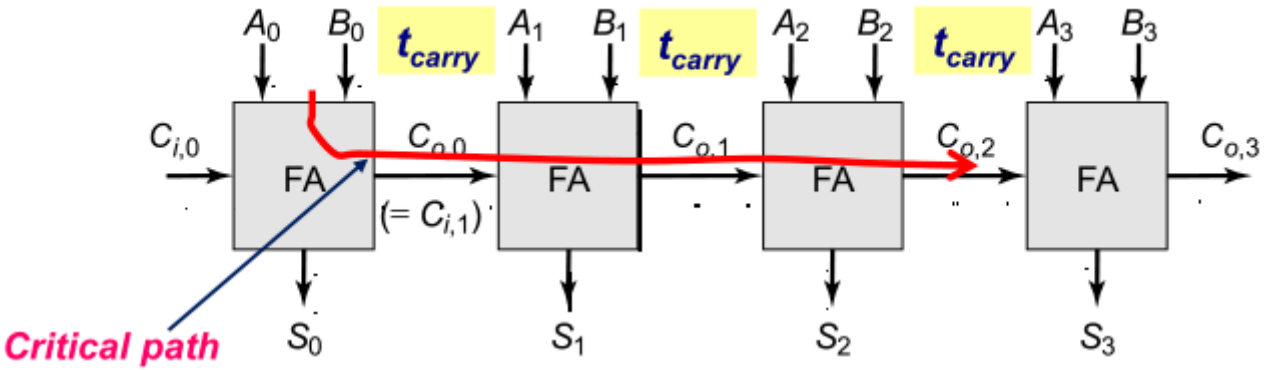
$$c_{out} = a \cdot b + a \cdot c_{in} + b \cdot c_{in}$$

$$s = a \oplus b \oplus c_{in} = a \cdot b \cdot c_{in} + \overline{c_{out}} \cdot (a + b + c_{in})$$

a	b	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

多位加法器结构

- Carry-Ripple Adder



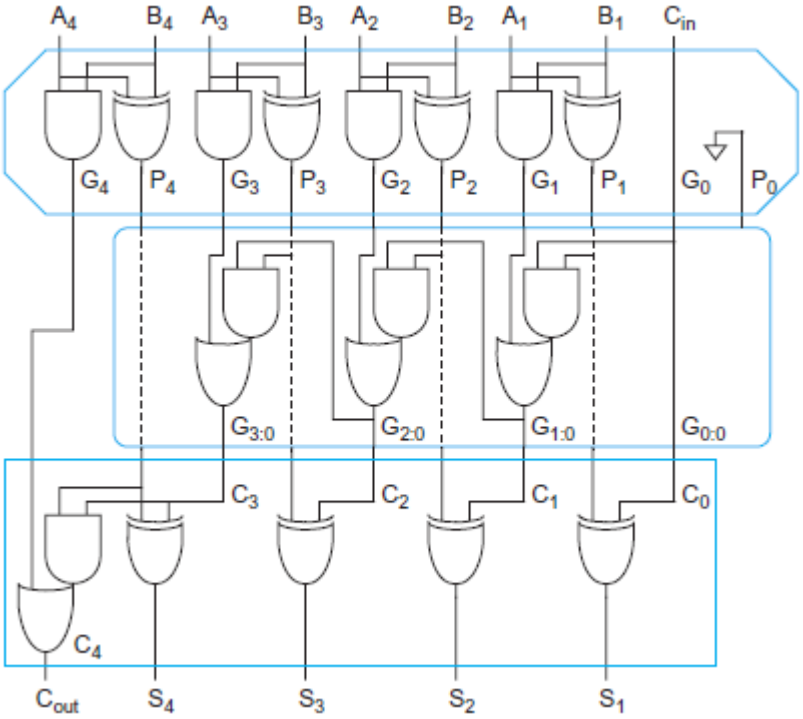
4-bit carry-ripple adder

Worst-case propagation delay:

$$t_{adder} = (N-1)t_{carry} + t_{sum}$$

多位加法器结构

GP Carry-Ripple Adder

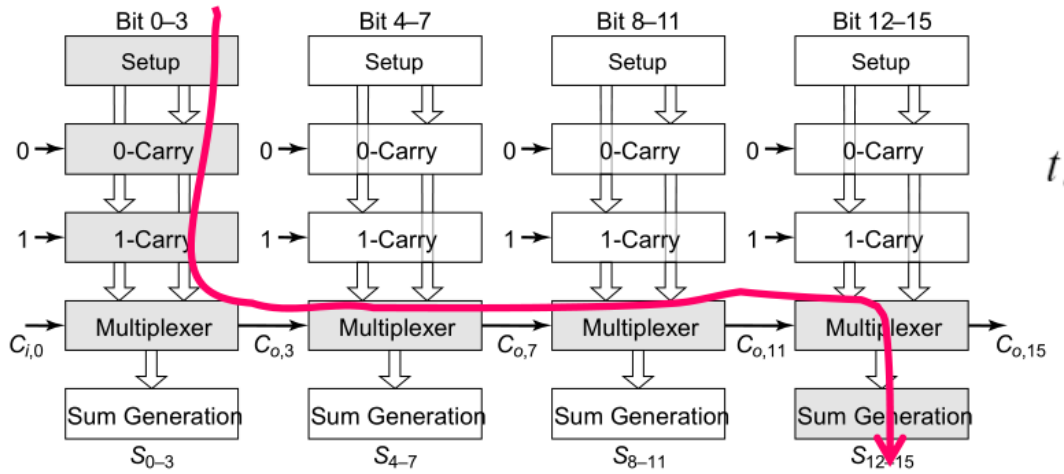


4-bit carry-ripple adder using PG logic

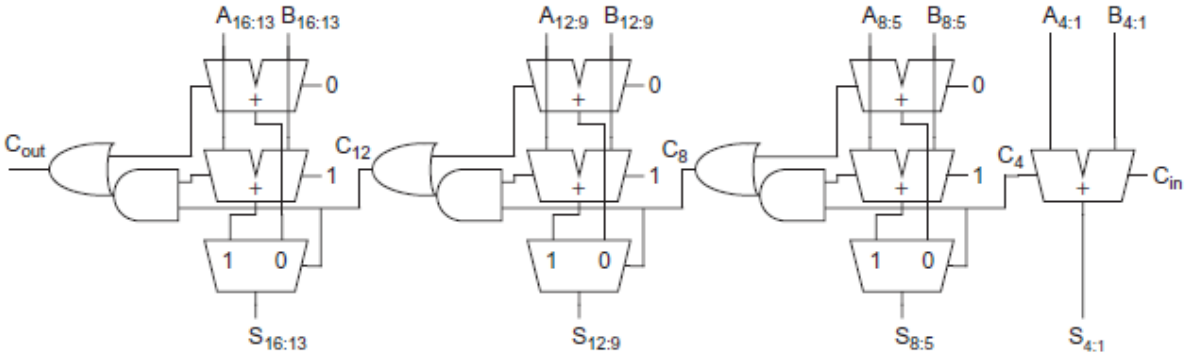
- $G_{n:0} = C_{o,n} = G_n + P_n \cdot C_{o,n-1}$
- $(G, P) \cdot (G', P') = (G + PG', PP')$
- $G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j}$
- $P_{i:j} = P_{i:k} \cdot P_{k-1:j}$
- $G_{i:i} = G_i = A_i \cdot B_i$
- $P_{i:i} = P_i = A_i \oplus B_i$
- $S_i = P_i \oplus G_{i-1:0}$
- $G_{0:0} = C_{in}$
- $P_{0:0}$=0

多位加法器结构

Carry-Select Adder

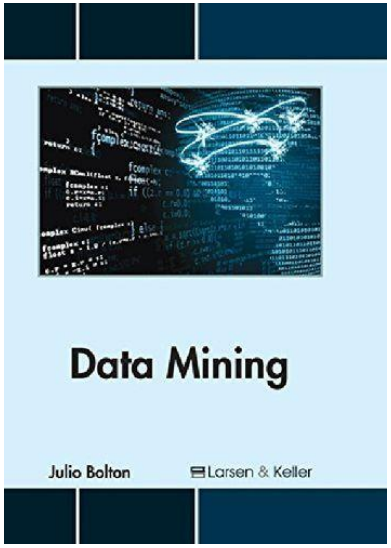


$$t_{add} = t_{setup} + M t_{carry} + \left(\frac{N}{M}\right) t_{mux} + t_{sum}$$



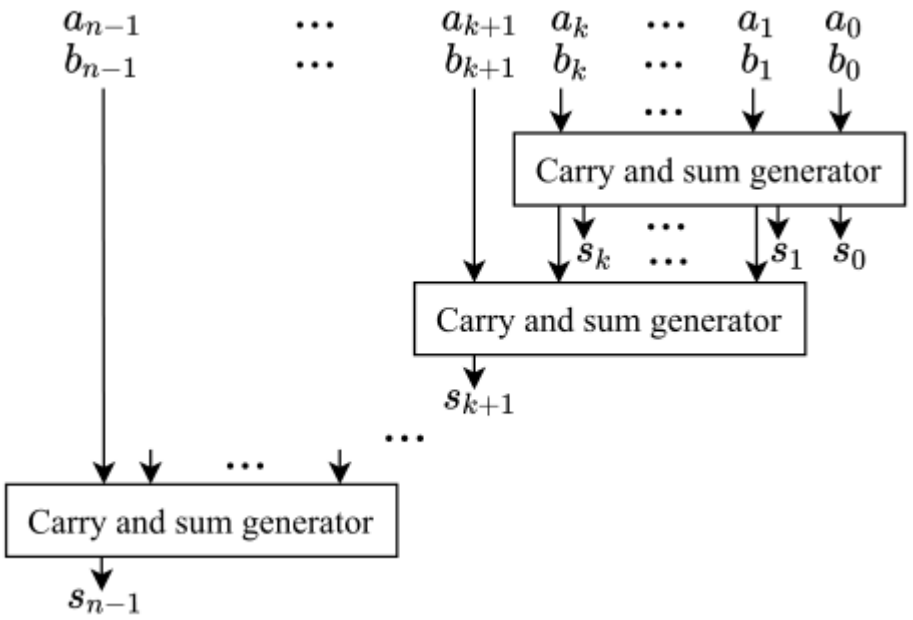
近似计算的应用需求

Error resilient



近似加法器原理

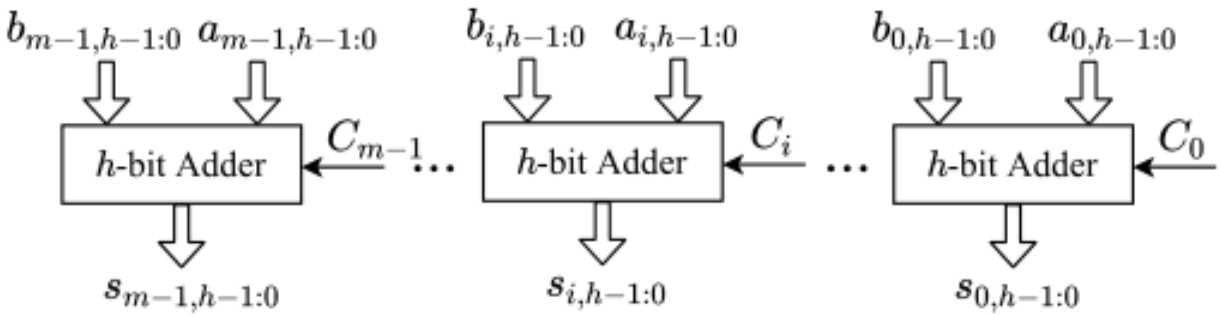
- Speculative Adder



an n-bit speculative adder uses the previous k bits (k < n) to predict the carry

近似加法器结构

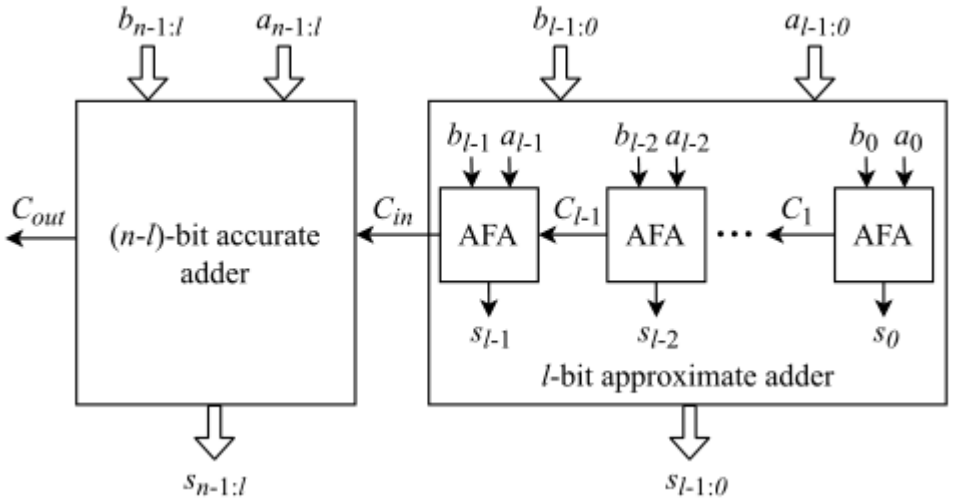
- Segmented Adder

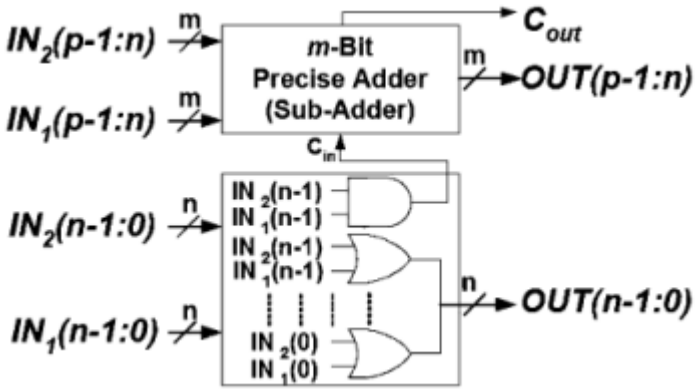


A segmented adder is implemented by several parallel subadder blocks with an independent carry-in

近似加法器结构

- Approximate Full Adder



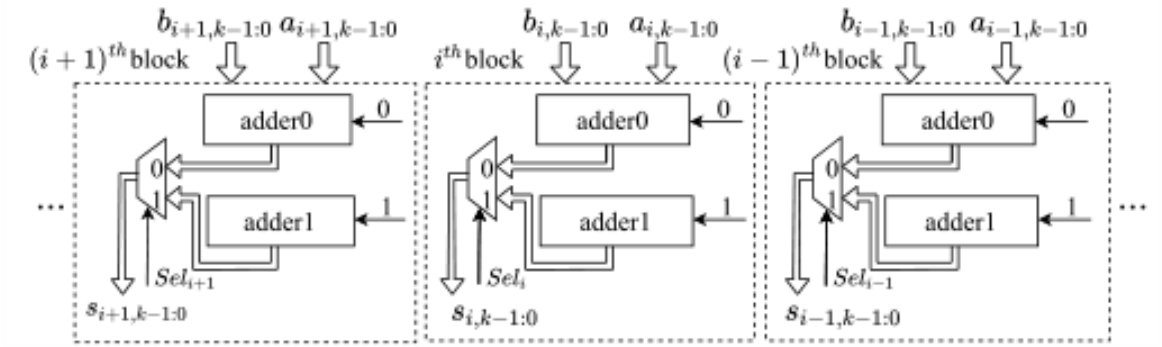


"Lower-Part-OR Adder(LOA)"

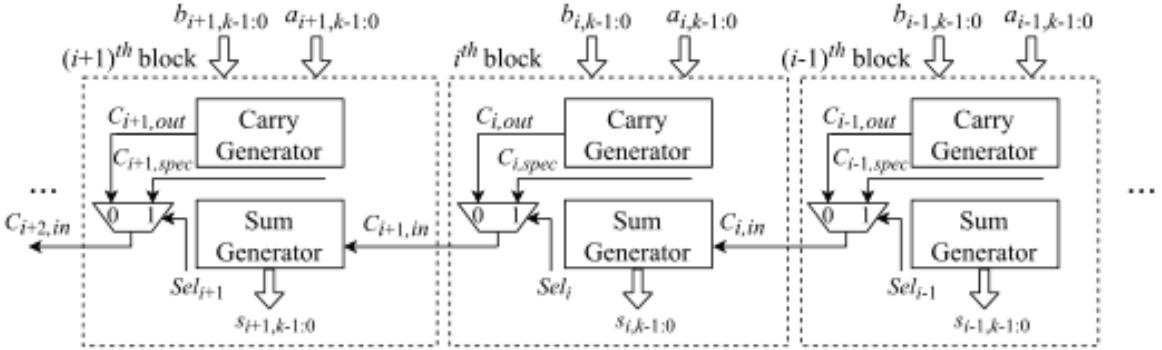
Approximate full adder is used to implement l LSBs in an n-bit adder (l < n), whereas the (n −l) MSBs are computed by an Accurate adder

近似加法器结构

- Approximate Carry-Select Adder



Sum selection



Carry-in selection

典型的近似加法器

- RAP-CLA
- ACAA
- GeAr
- HABA
- SARA
- RCPA
- ETA-I/ETA-II
- …

Outline

- 实验概述
  - 加法器简介
  - 近似计算加法器
- **实验要求**
  - 电路设计要求
  - 报告要求
- 附录

项目要求

- 项目目标1： **设计一个 32-bit 无符号加法器（作为baseline）**
  - 加法器结构可选择行波进位、超前进位等，也可根据文献选择其它结构；
  - 搭建 SPICE 网表仿真测试，得到关键路径延迟和功耗等信息；（对功耗的测量**自行选取测试向量***，提供了参考测试向量stimulus.vect见附件）

- 项目目标2： **设计一个 32-bit 无符号的低功耗近似计算加法器**
  - 自行查阅文献，选择合适（**错误率不高于 10%**）的近似加法器结构（在报告中指明设计结构及参考文献），搭建 SPICE 网表仿真测试；
  - 优化选取结构的**关键路径延迟和功耗**；
  - 与 baseline 进行功耗、延迟、漏电等各方面的比较，对近似加法器的优缺点进行讨论。（保持相同的测试向量）

*** 自选测试向量： 需要说明选取的合理性**

实验要求

需测量的参量

- **关键路径延时：**输入信号与输出信号50%翻转点间间隔（难点：**关键路径挑选，须在报告中述及**）
- **动态功耗：**基于自行选取测试向量测量平均动态功耗
- **泄漏功耗：**基于自行选取测试向量测量平均静态功耗
- **面积估计：**$\sum W \cdot L$或者粗略统计晶体管数量

实验要求

工艺

- 采用7nm PTM工艺库
- LP(low power)

仿真环境

- **输入信号上升下降时间：**15ps
- **额定电压：**0.7V
- **温度：**25 ℃
- **输出端口负载：**8 倍最小尺寸 inverter 门电容负载
- **命名规则：**输入与输出分别为A[31:0], B[31:0], SUM[32:0]

报告事宜

- 报告内容：
  - 前期调研结果简述（结构选择，逻辑方式选择），明确设计目标和设计策略的选用
  - SPICE网表描述，关键路径及data pattern选择
  - 逻辑组织框图（标明关键路径）
  - 优化设计方案（如sizing方案，设计技巧）
  - 性能、功耗仿真结果分析
  - 报告最后将电路网表贴出
- 提交方式：
  - Canvas 平台
  - Deadline：2022.x.xx

Outline

- 实验概览
  - 加法器简介
  - 近似计算加法器设计
- 实验要求
  - 电路设计要求
  - 报告要求
- **附录**

向量激励定义

Digital vector file commands

- IO Syntax: IO I | O | B | U (信号方向输入、输出、双向、无用)
- VNAME：Syntax: VNAME vector_name[[starting_index:ending_index]](定义向量名字)
- RADIX: Syntax: RADIX NUMBER_OF_BITS(定义每个向量bit数)
- PERIOD：Syntax: PERIOD TIME_INTERVAL(定义数据表中数据间隔)
- TUNIT: Syntax:TUNIT [fs|ps|ns|us|ms]（定义VEC文件中时间单位）
- TEDLAY Syntax: TDELAY DELAY_VALUE [MASK]（定义数据表中输入输出延时）
- VIL/VIH/VOH/VOL/VTH Syntax: keywords logic_voltage [mask]（定义逻辑输入/输出低/高电平，开关阈值）
- TFALL/TRISE/SLOPE：Syntax:keywords [trise|tfall] [mask]（定义输入信号斜率)

向量激励例子

Digital Vector File Example

```
; specifies # of bits associated with each vector
radix 1 2 444
;*******************************************************
; defines name for each vector. For multi-bit vectors,
; innermost [] provide the bit index range, MSB:LSB
vname v1 va[[1:0]] vb[12:1]
;actual signal names: v1, va[0], va[1], vb1, vb2, ... vb12
;*******************************************************
; defines vector as input, output, or bi-directional
io i o bbb
; defines time unit
tunit ns
;*******************************************************
; vb12-vb5 are output when 'v1' is 'high'
enable v1 0 0 FF0
; vb4-vb1 are output when 'v1' is 'low'
enable ~v1 0 0 00F
;*******************************************************
; all signals have a delay of 1 ns
; Note: do not put the unit (such as ns) here again.
; HSPICE multiplies this value by the specified 'tunit'.
tdelay 1.0
; va[1] and va[0] signals have 1.5ns delays
tdelay 1.5 0 3 000
;*******************************************************
; specify input rise/fall times (if you want different
; rise/fall times, use the trise/tfall statement.)
; Note: do not put the unit (such as ns) here again.
; HSPICE multiplies this value by the specified 'tunit'.
slope 1.2
;*******************************************************
; specify the logic 'high' voltage for input signals
vih 3.3 1 0 000
vih 5.0 0 0 FFF
; to specify logic low, use 'vil'
;*******************************************************
; va & vb switch from 'lo' to 'hi' at 1.75 volts
vth 1.75 0 1 FFF

;*******************************************************
; tabular data section
10.0 1 3 FFF
20.0 0 2 AFF
30.0 1 0 888
```

施加测试向量：

1）上述文件保存为stimulus.vect

2）在testbench.sp中添加仿激励文件：.vec stimulus.vect

3）Testbench中对应的向量的激励即stimulus.vect中定义的数据表

参考文献

- **TCASI 2018**，*Design and Evaluation of Approximate Logarithmic Multipliers for Low Power Error-Tolerant Applications*
- **VLSI 2018**，*Systematic Design of an Approximate Adder The Optimized Lower Part Constant-OR Adder*
- **TCASII 2018**，*RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder*
- **VLSI 2020**，*An Efficient Parallel DA-Based Fixed Width Design for Approximate Inner-Product Computation*
- **TCASII 2020**，*Stochastic Mixed-PR A Stochastically-Tunable Low-Error Adder*
- **TCASI 2021**，*Accuracy-Configurable Radix-4 Adder With a Dynamic Output Modification Scheme*