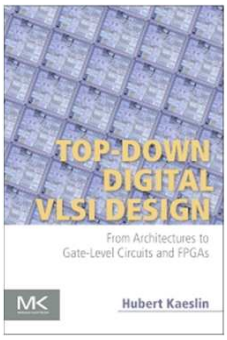
ECE4740: Digital VLSI Design

Lecture 26: Architecture transforms

954

Recap

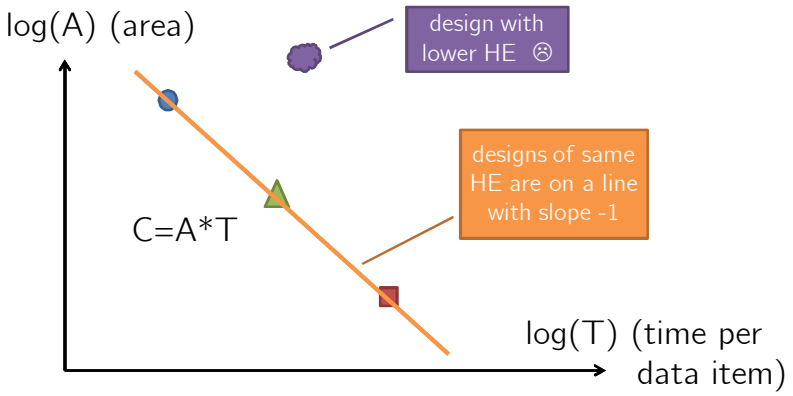
Architecture transforms*



*H. Kaeslin, "Digital Integrated Circuit Design," Cambridge Univ. Press, 2008

955

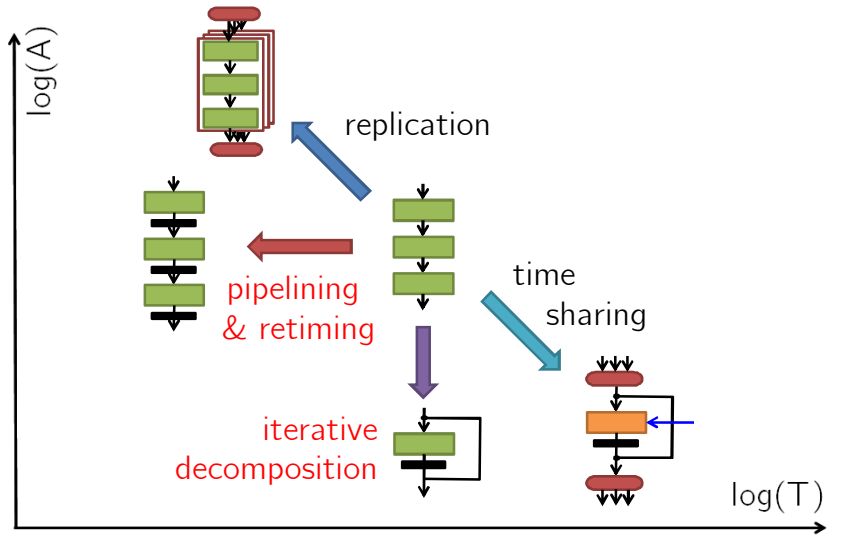
AT diagram



- Hardware efficiency: HE = A*T
- Helpful to compare different designs
- Visualize architecture transforms

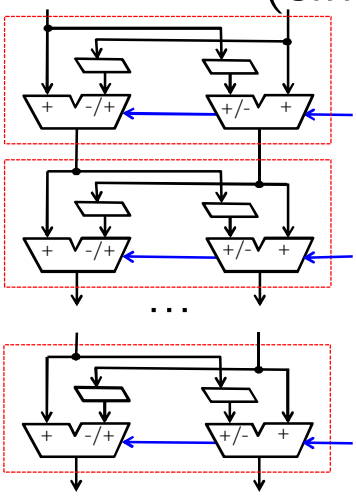
956

Architecture transforms



957

Recall: rotation CORDIC (simplified)

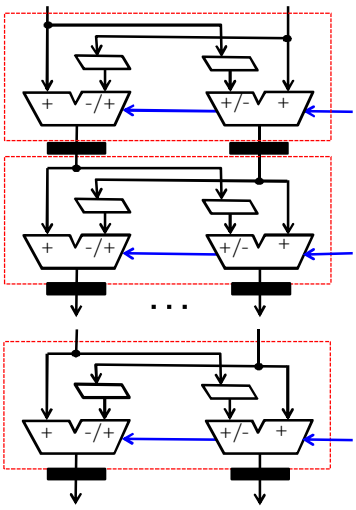


- Isomorphic architecture:
  - Each operation has its own circuit
- $A_{iso}=K*A_{PR}$
- $T_{iso}=K*T_{PR}$

PR = pseudo rotation

958

Fully pipelined architecture



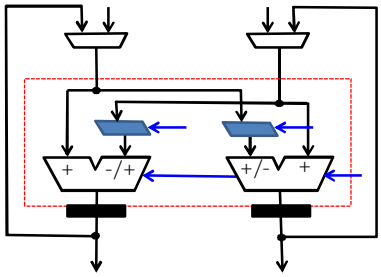
- Pipelined architecture
- $A_p=K*A_{PR}+K*A_{ff}$
- $T_p=T_{PR}+T_{ff}$

includes propagation delay and setup time

- How much is the latency?

959

Fully decomposed architecture



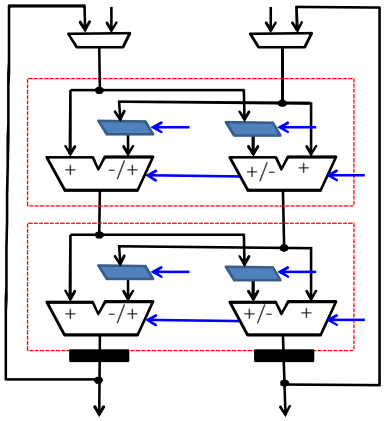
- Architecture needs MUX at input and configurable shifter
- $A_u = A_{PR} + A_{ff} + A_{MUX}$
- $T_u = K*(T_{PR} + T_{ff} + T_{MUX})$

note that shifter must be programmable and hence will be more complex

- How much is the latency?

960

2x decomposed architecture



- Partially unrolling the architecture
- $A_{u2} = 2(A_{PR} + A_{ff} + A_{MUX})$
- $T_{u2} = ½K(T_{PR} + T_{ff} + ½T_{MUX})$

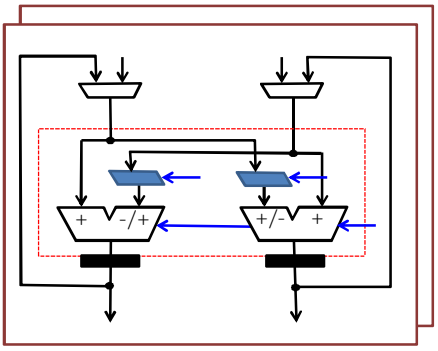
less impact of the multiplexer

- There is a HE-optimal unroll factor!

*shifters now only support 1/2 of the modes

961

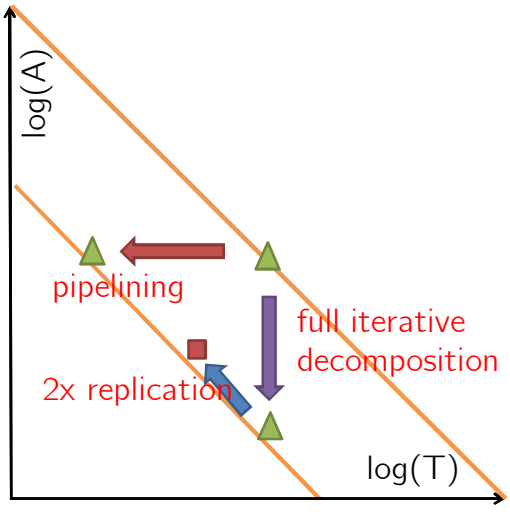
2x replicated architecture



- Replication makes things 2x faster at 2x the area
- A_{ur}= $2*(A_{PR}+A_{ff}+A_{MUX})$
- T_{ur}= $K/2*(T_{PR}+T_{ff}+T_{MUX})$

962

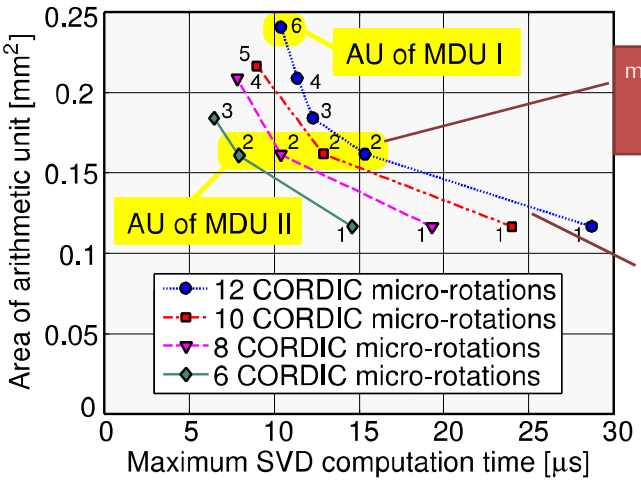
Example of CORDIC transforms



- Optimized designs will have similar HE
- Optimal design depends on the throughput requirements or area constraints

963

Real-world CORDIC trade-off



micro-rotations per cycle = unroll factor

non-idealities reduce HE

180nm ASIC

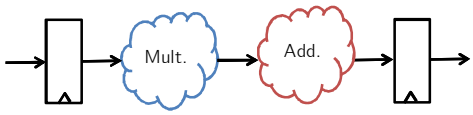
CS, P. Blösch, P. Friedli, and A. Burg, "Matrix Decomposition Architecture for MIMO Systems: Design and Implementation Trade-offs," ASILOMAR, 2007

Architecture transforms summary

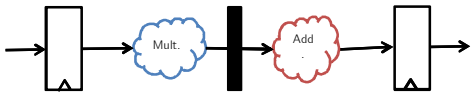
- General tool to optimize hardware-efficiency or to meet target specifications:
  - Throughput
  - Area
  - (Power?)
- First-order estimates area and delay can be obtained with hand calculations
  - Non-idealities or secondary effects determine real A and T

965

Pipelining can reduce area

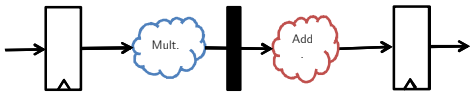


- Assume tight timing constraint so that fast and large adder and multiplier must be used
- Pipelining relaxes timing and may enable the use of slower but smaller circuits



966

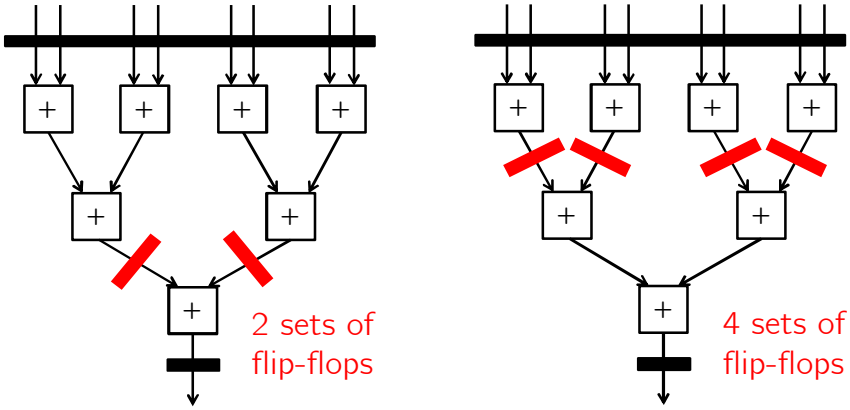
Pipelining can reduce power



- Lower area typically leads to lower power
- Pipeline stages stop glitch propagation
  - Lower dynamic power consumption
- Pipelining increases clock frequency
  - Make circuit faster than required and apply voltage-frequency scaling to reduce power
- Overhead of flip-flops may increase power

967

Retiming can increase the area



968

Horner's method

Practice architecture transforms

969

Your turn: Horner's method

- Evaluate polynomial:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

- Straightforward approach requires 4 additions and 10 multiplications
- Horner's method:

$$f(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + x(a_4))))$$

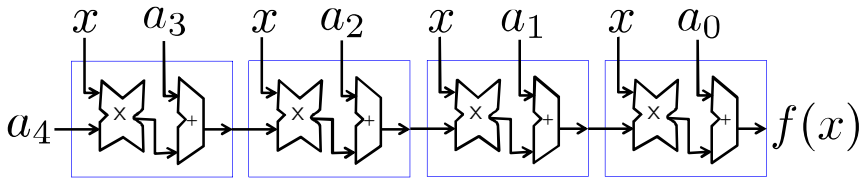
requires only 4 multiplications*

*again, algorithmic transforms help the most!

970

Your turn: Horner's method (cont'd)

- Isomorphic VLSI architecture:

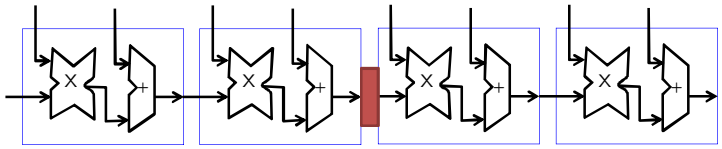


  - Assume a_{MAC}=10 and t_{MAC}=10 for MAC unit
  - Assume a_{ff}=1 and t_{ff}=1 for flip-flops
- Perform pipelining (add 1 & 3 stages)
- Compute A, T and HE

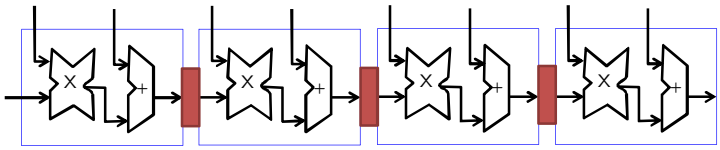
971

Architectures*

- 1x pipelined: A=40+1, T=20+1, AT=861



- 3x pipelined: A=40+3, T=10+1, AT=473

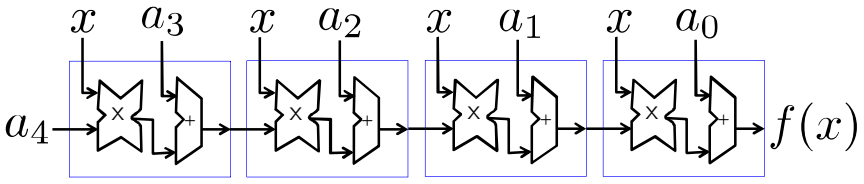


*ignoring registers used to align latency!!!

972

Your turn: Horner's method (cont'd)

- Isomorphic VLSI architecture:

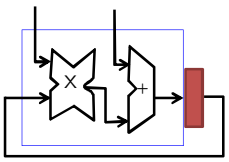


  - Assume a_{MAC}=10 and t_{MAC}=10 for MAC unit
  - Assume a_{ff}=1 and t_{ff}=1 for flip-flops
- Perform iterative decomposition (1 & 2 stages per cycle)
- Compute A, T and HE

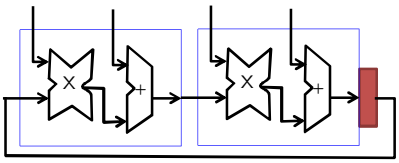
973

Architectures* (cont'd)

- Full iterative decomposition: A=10+1, T=4(10+1), AT=484



- Iteratively decomposed: A=20+1, T=2(20+1), AT=882

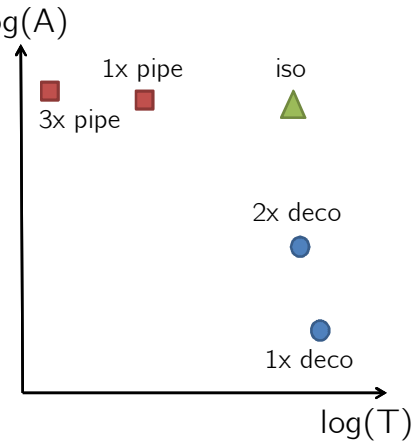


*ignoring latency alignment and coefficient registers!

974

Trade-offs for Horner's method

Architecture	Area	Time	A*T
Isomorphic	40	40	1600
1x pipelined	41	21	861
3x pipelined	43	11	473
1x decomposed	11	44	484
2x decomposed	21	42	882
Original formulation*	~100	~80	~8000



*estimates for original polynomial formulation

975