Area Minimization of Redundant CORDIC Pipeline Architectures

A.Wassatsch, S.Dolling, D.Timmermann

ICCD'98

Austin,Texas(USA)

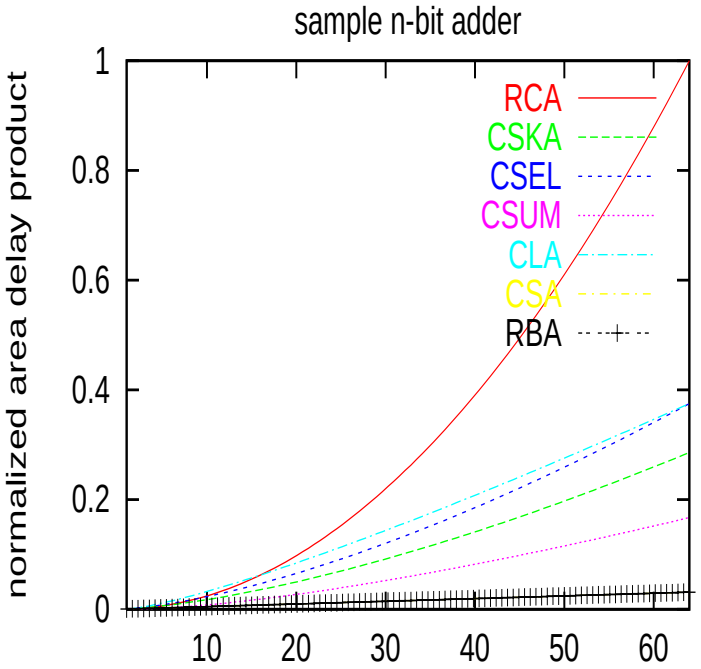
October 5, 1998

University Rostock, Germany
Department of Electrical Engineering and IT
Institute of Applied Microelectronics and CS

Outline

- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- Area reduction method for add&shift algorithms
- Application for CORDIC
- Benefits of area reduction
- Conclusion

Motivation



area-delay product vs. wordlength

- redundant arithmetic
  - fast, delay independent of wordlength, but *very chip area consuming*
- observation
  - similarity of result generation to digit-on-line algorithms (MSD-first)
  - investigation of behavior of transfer digits in redundant adder arrays

Outline

- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- Area reduction method for add&shift algorithms
- Application to CORDIC
- Benefits of area reduction
- Conclusion

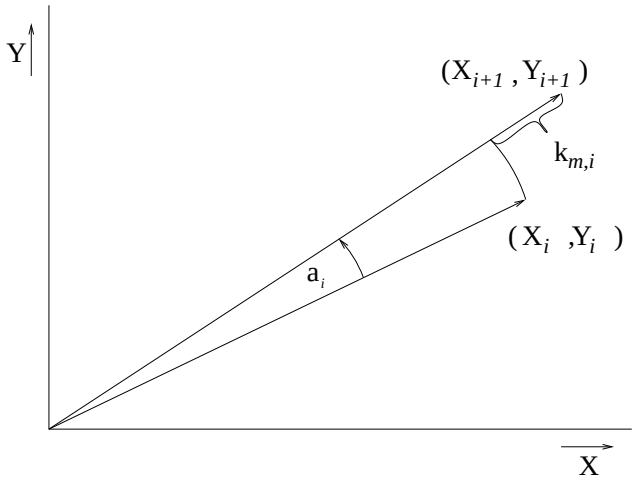
Introduction to CORDIC algorithm

	rotation ($z_n \to 0$)	vectoring ($y_n \to 0$)
trigonometric ($m = 1$)	$x_n = k_1(x_0 \cos(z_0) - y_0 \sin(z_0))$ $y_n = k_1(y_0 \cos(z_0) + x_0 \sin(z_0))$	$x_n = k_1 \sqrt{x_0^2 + y_0^2}$ $z_n = z_0 + \tan^{-1}(y_0/x_0)$
linear ($m = 0$)	$x_n = x_0$ $y_n = x_0 z_0 + y_0$	$x_n = x_0$ $z_n = z_0 + y_0/x_0$
hyperbolic ($m = -1$)	$x_n = k_{-1}(x_0 \cosh(z_0) + y_0 \sinh(z_0))$ $y_n = k_{-1}(y_0 \cosh(z_0) + x_0 \sinh(z_0))$	$x_n = k_{-1} \sqrt{x_0^2 - y_0^2}$ $z_n = z_0 + \tanh^{-1}(y_0/x_0)$

Introduction to CORDIC algorithm (cont')

iteration:

$$
\begin{aligned}
x_{i+1} &= x_i - m\sigma_i 2^{-S(m,i)} y_i \\
y_{i+1} &= y_i + \sigma_i 2^{-S(m,i)} x_i \\
z_{i+1} &= z_i - \sigma_i \alpha_{m,i}
\end{aligned}
$$



scaling:

$$
\begin{aligned}
x &= x_n k_m^{-1} \\
y &= y_n k_m^{-1}
\end{aligned}
$$

with

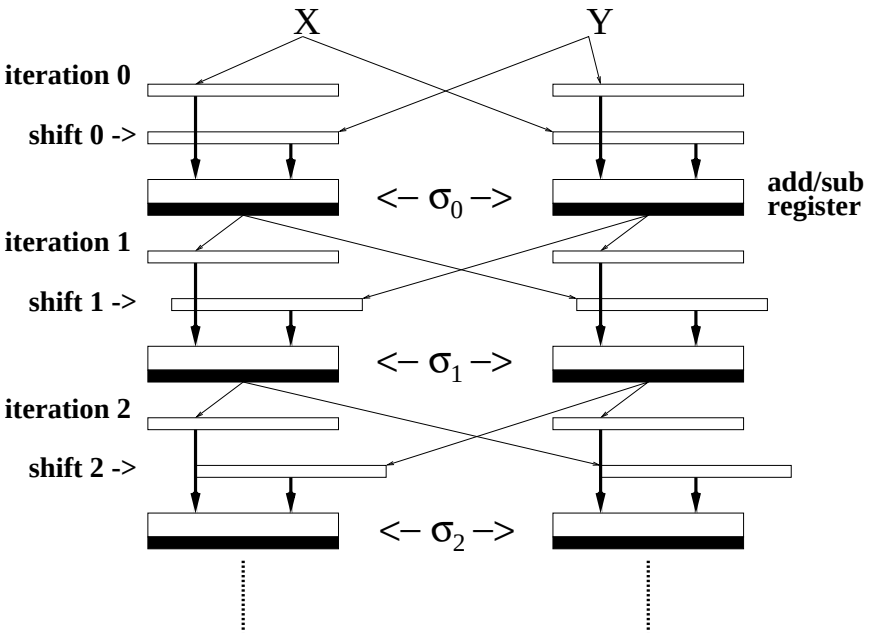
$$
k_m = \prod_{i=0}^{n-1} k_{m,i} = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + m\sigma_{m,i}^2}}
$$

$$
\alpha_m = \sum_{i=0}^{n-1} \sigma_i \alpha_{m,i} = \sum_{i=0}^{n-1} \frac{\sigma_i \tan^{-1}(\sqrt{m}\sigma_{m,i})}{\sqrt{m}}
$$

$\alpha_{m,i}$: **rotation angle**

σ_i : **rotation direction**

Introduction to CORDIC algorithm (cont')



principle of add&shift for CORDIC pipeline
(z-datapath not shown)

- based on add & shift operations
- build a regular array
- intense communication between two of the three datapaths

Outline

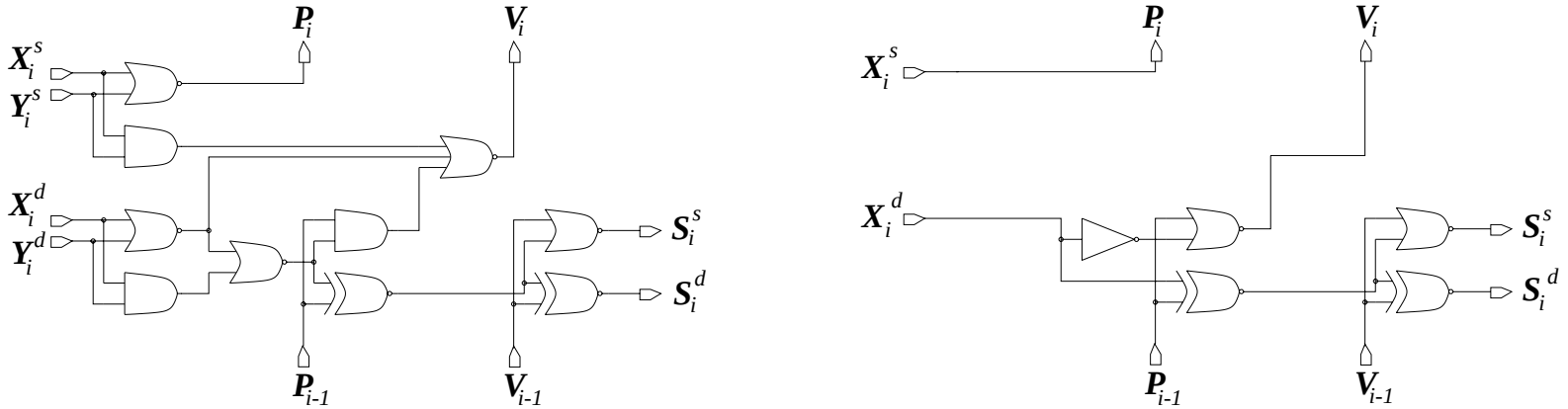
- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- Area reduction method for add&shift algorithms
- Application to CORDIC
- Benefits of area reduction
- Conclusion

Previous approaches

- area reduction by algorithmic modifications
  - integration of scaling into the iteration; optimization of the special scaling operation [Schmidt, et al., 1986]
  - σ_i estimation [Takagi et al., 1991], [Lee and Lang, 1992]
  - reducing the number of iteration repetitions [Timmermann et al., 1992]
  - booth recoding of σ_i [Timmermann et al., 1992], [Antelo et al., 1996]

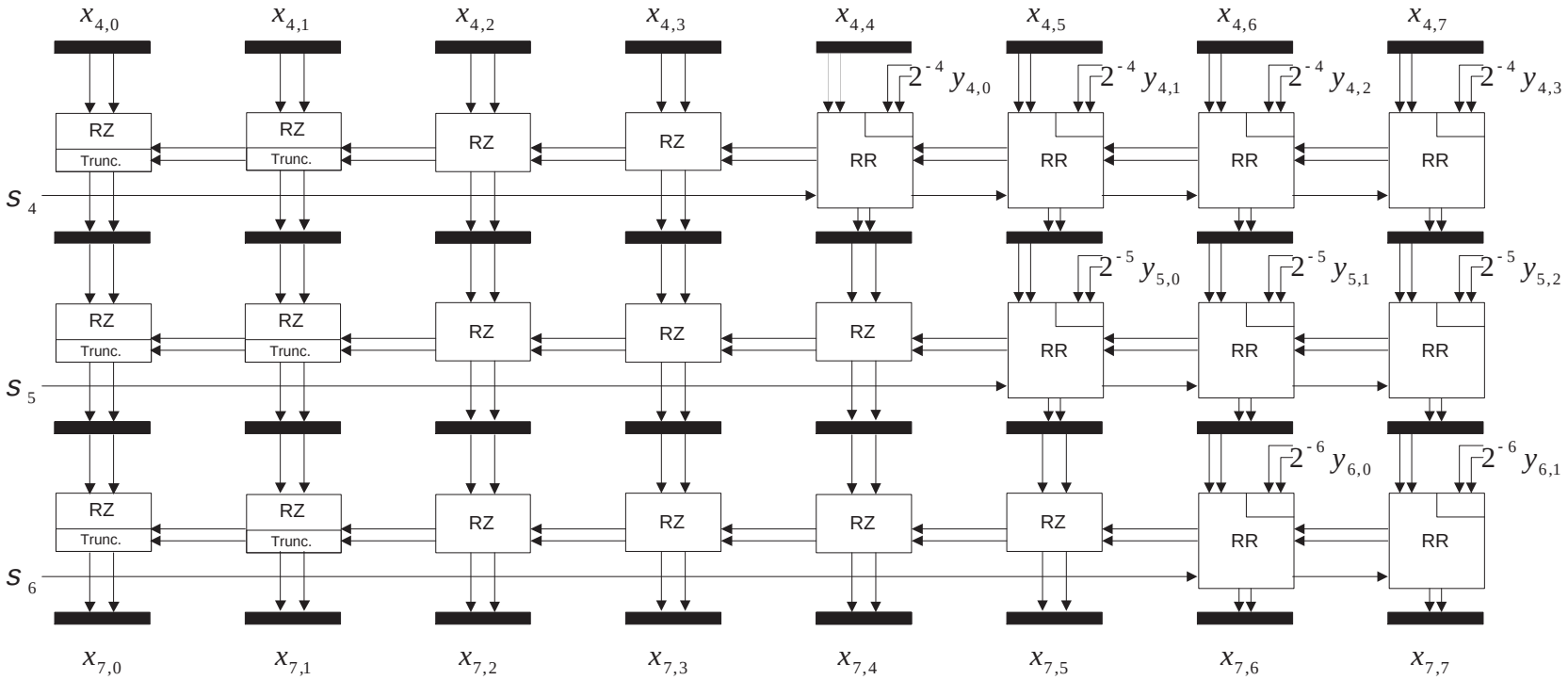
Previous approaches (cont')

- area reduction on bit-level
  - nonredundant architectures
    [Timmermann and Sundsbø, 1992]
  - redundant-zero adder utilizes increasing shifts



Redundant adder-cell (RR) $\longrightarrow$ Redundant zero adder-cell (RZ)

Previous approaches (cont')

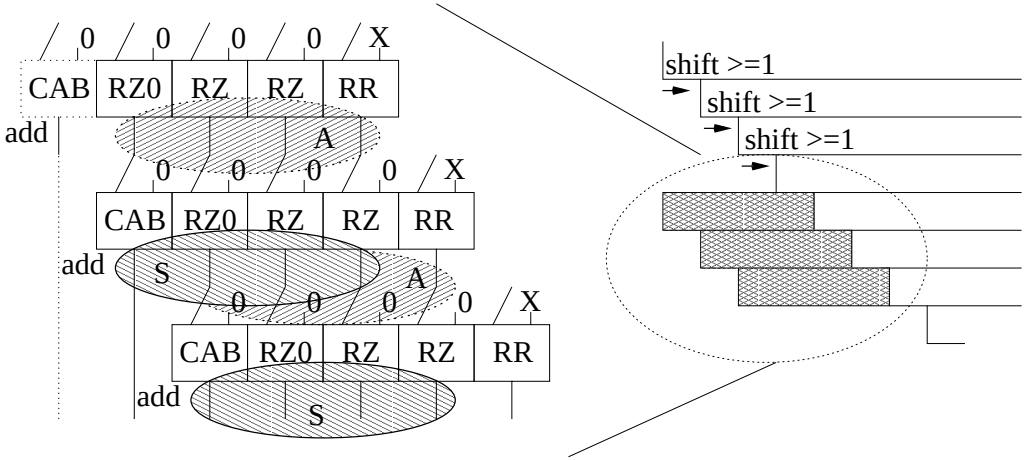


Reducing x-datapath using RZ-cells

Outline

- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- **Area reduction method for add&shift algorithms**
- Application to CORDIC
- Benefits of area reduction
- Conclusion

A closer look at redundant addition



$$S = A + B + c_{in}$$

$$s_i, a_i \in \{\bar{1}, 0, 1\}$$

$$b_i = 0$$

$$c_{in} \in \{\bar{1}, 0, 1\}$$

$$i = 0, 1, 2, 3$$

$\{(a_3, a_2, a_1), (a_2, a_1, a_0)\} \neq \{(111), (\bar{1}\bar{1}\bar{1})\}$

$-13 \leq A \leq 13 \rightarrow -14 \leq S \leq 14$

$s_4 \neq 0$ with recoding $\bar{1}1 \rightarrow 0\bar{1}$, $1\bar{1} \rightarrow 01$, $\bar{1}01 \rightarrow 0\bar{1}\bar{1}$, $10\bar{1} \rightarrow 011$

results in $s_4 = 0$ and $\{(s_3, s_2, s_1), (s_2, s_1, s_0)\} \neq \{(111), (\bar{1}\bar{1}\bar{1})\}$

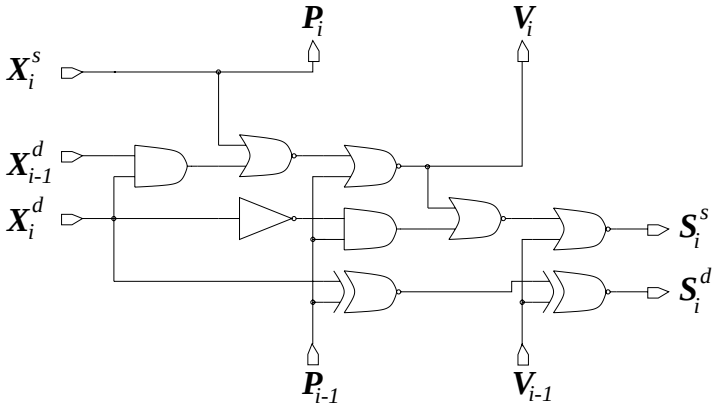
Redundant addition of leading zero's

		$a_{0,n-1}$	$a_{0,n-2}$	$a_{0,n-3}$	$a_{0,n-4}$	$a_{0,n-5}$	$a_{0,n-6}$	$a_{0,n-7}$
$+$		0	0	0	0	$b_{0,n-1}$	$b_{0,n-2}$	$b_{0,n-3}$
	s_4	s_3	s_2	s_1	s_0			
		$\mathbf{a_{1,n-1}}$	$a_{1,n-2}$	$a_{1,n-3}$	$a_{1,n-4}$	$a_{1,n-5}$	$a_{1,n-6}$	$a_{1,n-7}$
$+$			0	0	0	0	$b_{1,n-1}$	$b_{1,n-2}$
		s_4	s_3	s_2	s_1	s_0		
		$\mathbf{a_{1,n-1}}$	$\mathbf{a_{2,n-2}}$	$a_{2,n-3}$	$a_{2,n-4}$	$a_{2,n-5}$	$a_{2,n-6}$	$a_{2,n-7}$
$+$				0	0	0	0	$b_{2,n-1}$
			s_4	s_3	s_2	s_1	s_0	
		$\mathbf{a_{1,n-1}}$	$\mathbf{a_{2,n-2}}$	$\mathbf{a_{3,n-3}}$	$a_{3,n-4}$	$a_{3,n-5}$	$a_{3,n-6}$	$a_{3,n-7}$

(**bold face** = fixed values)

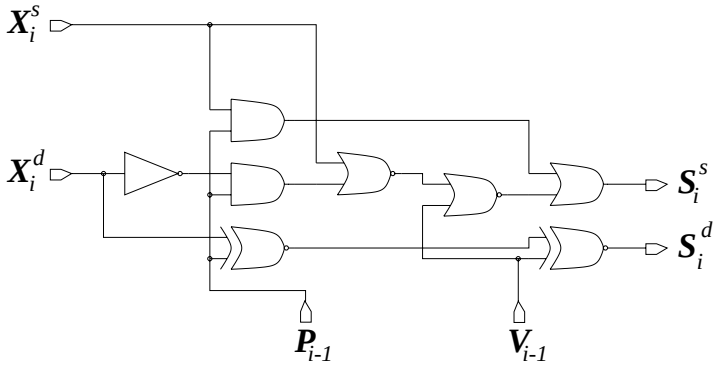
New cells for area reduction

- suppression of pseudo-overflows



Redundant zero 0 cell (RZ0)

- absorbs any possible carry, stops the flow of transfer digits P and V



Carry absorber cell (CAB)

Outline

- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- Area reduction method for add&shift algorithms
- Application to CORDIC
- Benefits of area reduction
- Conclusion

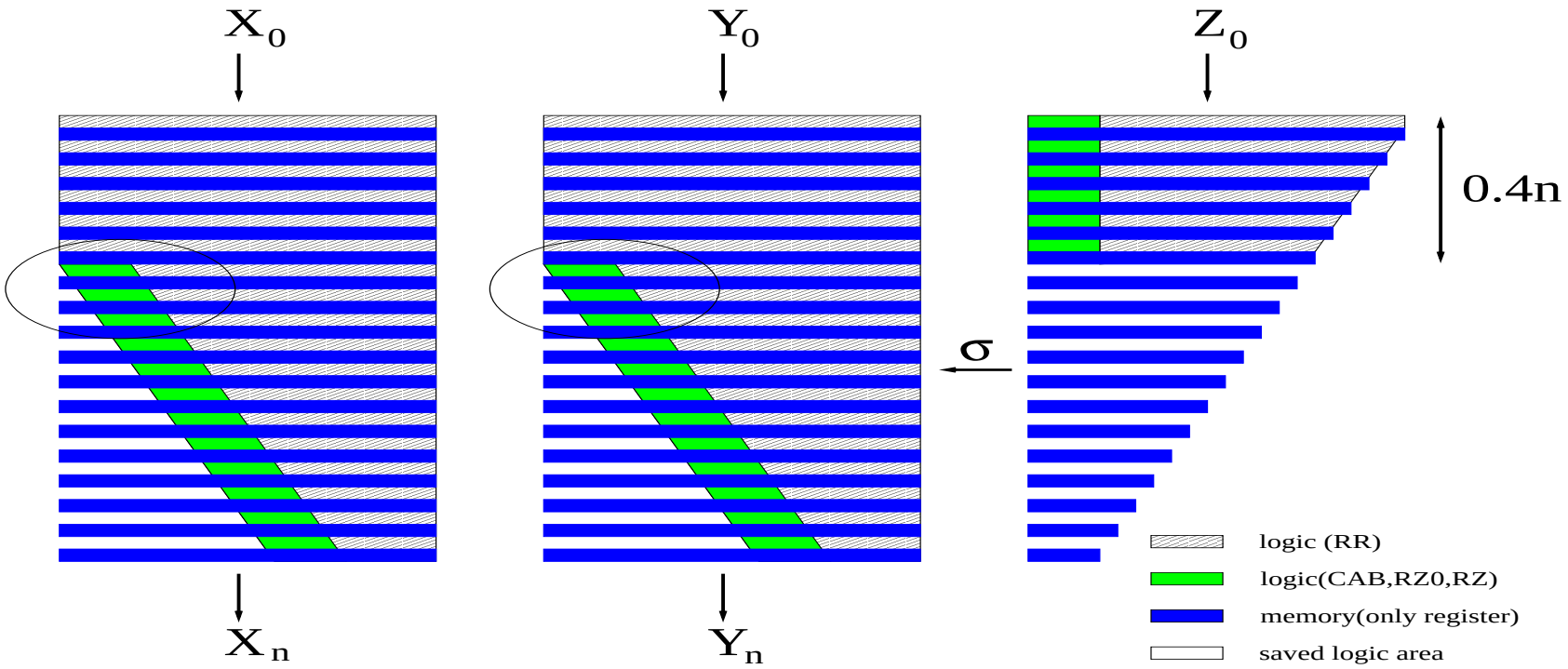
Area reduction application: CORDIC rotation mode

- reduction depends on operation mode, different implementation for each mode
- due to CORDIC-specific double iteration delayed start of reduction method
- two alternative implementation possible
  - CAB-RZ0-RZ-RZ
  - CAB-REC

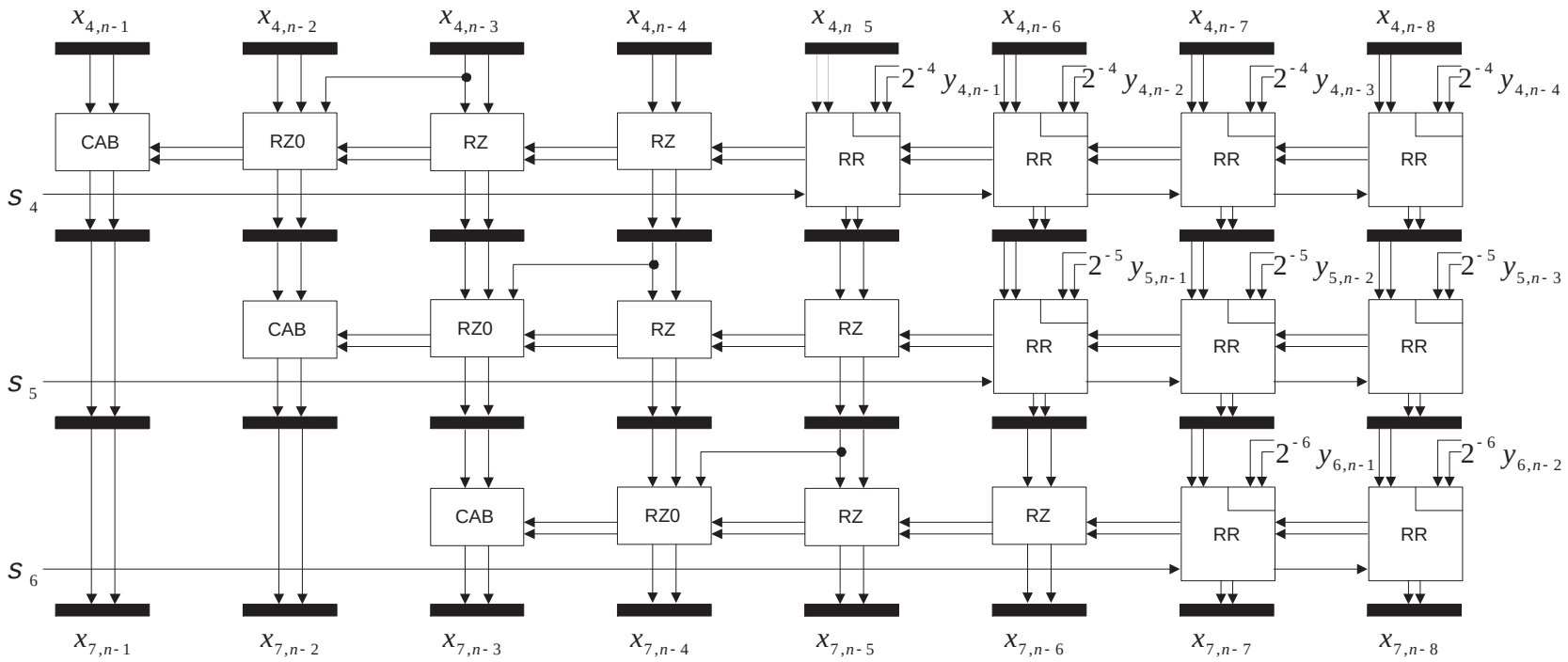
    advantage: starts one iteration before

    disadvantage: small increase in computing time (technology dependent)

Chip area reduction rotation mode CORDIC

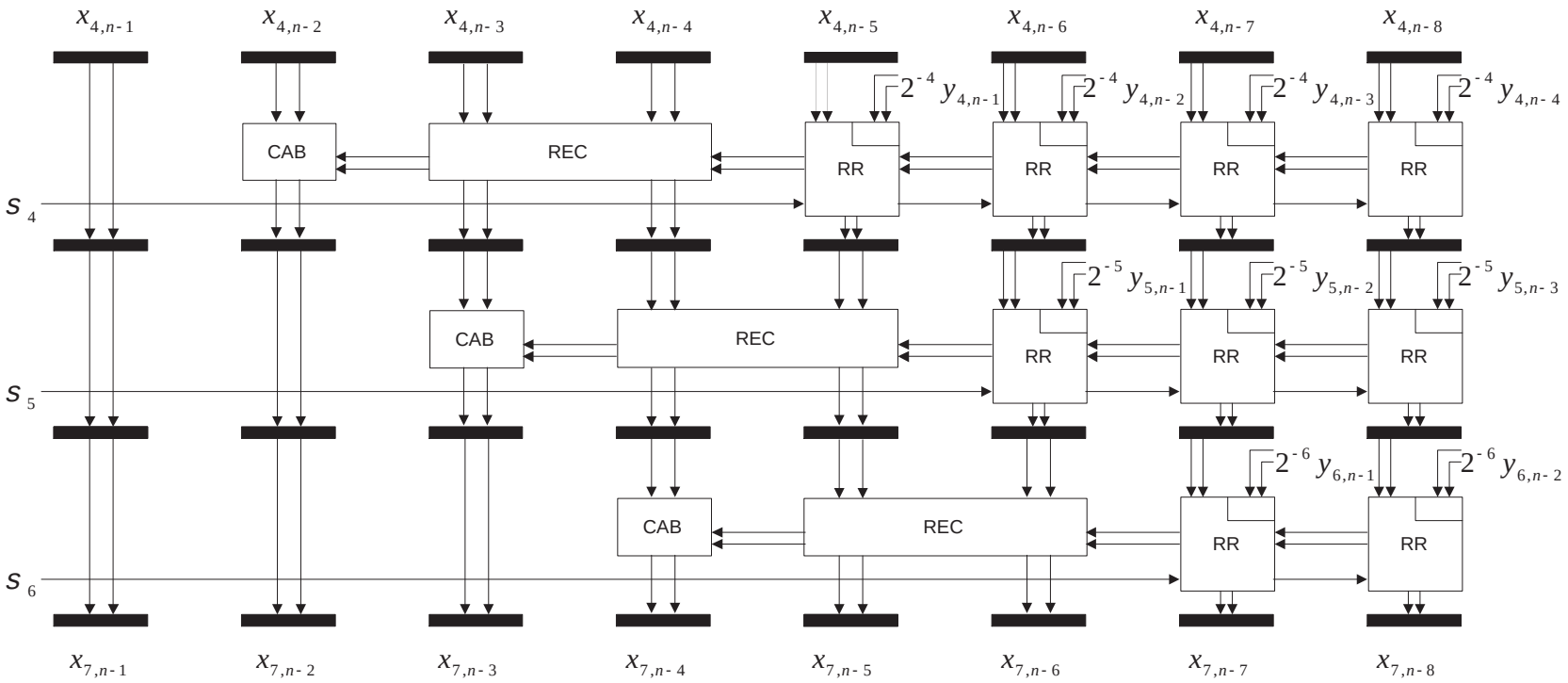


Area reduced datapath



Novel area reduced x-datapath by using special adder cells

Area reduced datapath, second version



Novel area reduced x-datapath by using 3 digit adder cells

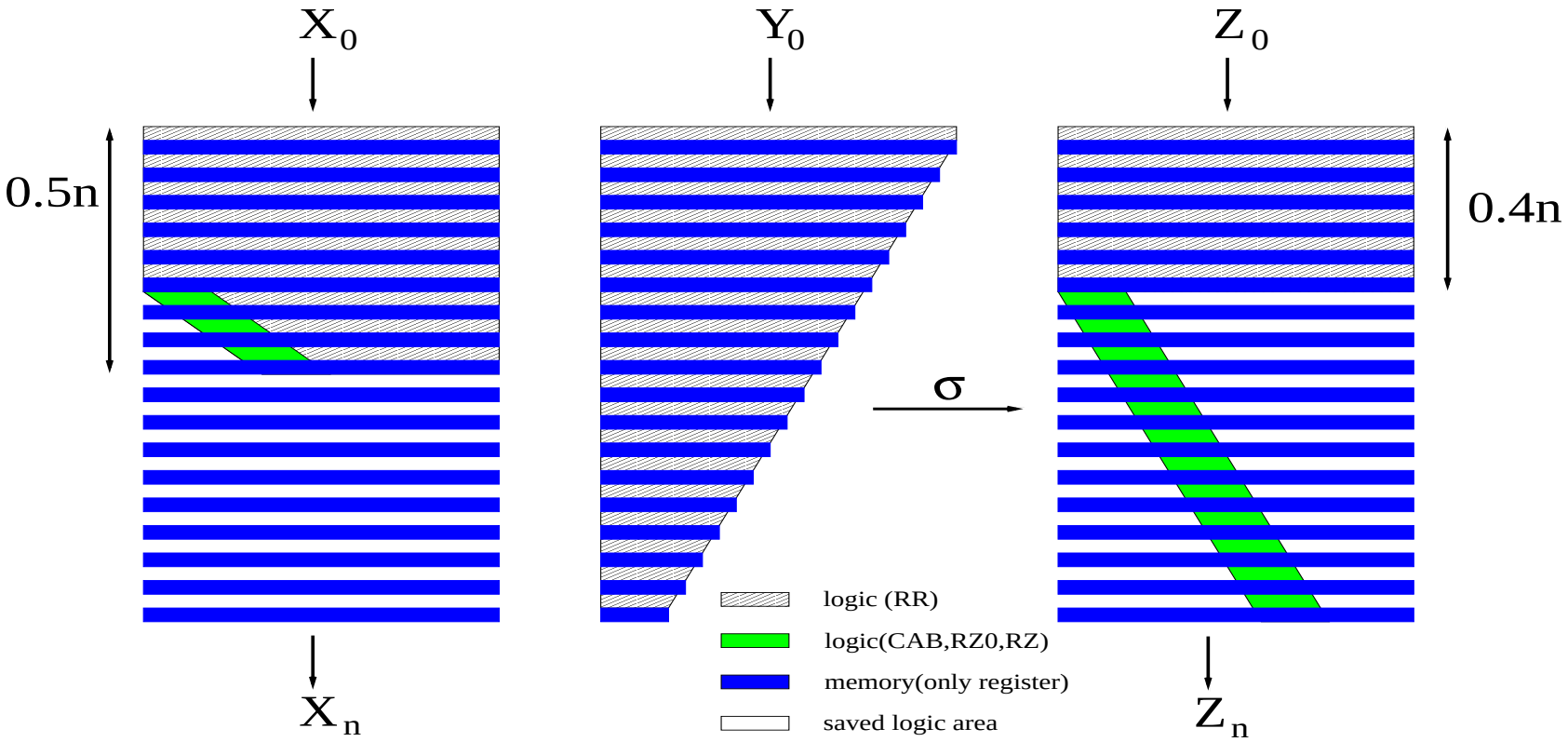
Area reduction for CORDIC vectoring mode

- modified iteration

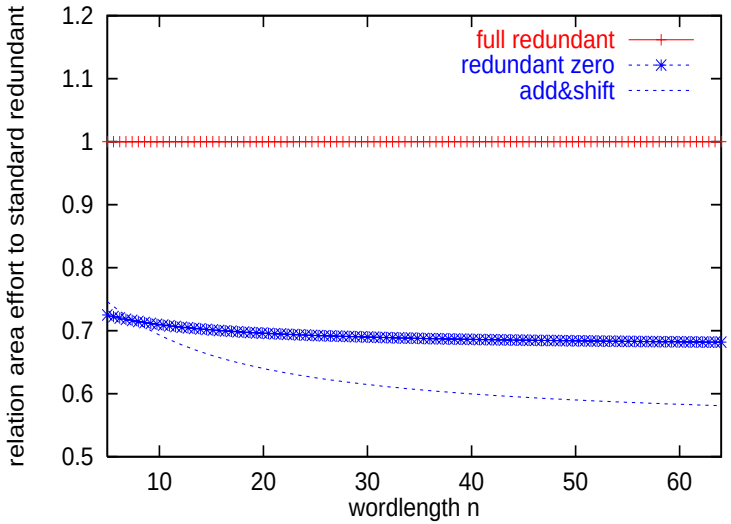
$$\begin{aligned} x_{i+1} &= x_i - m\sigma_i 2^{-2S(m,i)} y_i \\ y_{i+1} &= 2(y_i + \sigma_i x_i) \\ z_{i+1} &= z_i - \sigma_i \alpha_{m,i} \end{aligned}$$

- larger hardware savings due to larger right shift in x-datapath
- only registers for x required after iteration $i \geq \lceil n/2 \rceil$
- y resembles to the situation in the z-path for rotation mode
- only a small strip of special cells in the z-path after $\lceil n/3 \rceil$

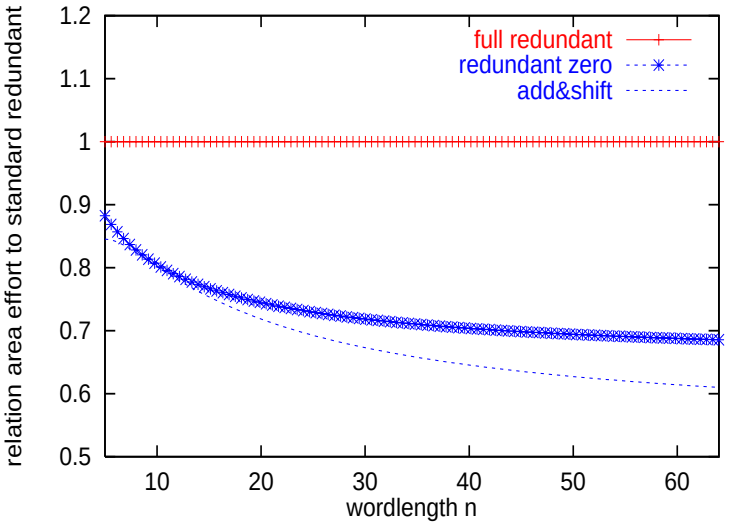
Chip area reduction vectoring mode CORDIC



Area comparison of redundant CORDIC architectures



rotation mode

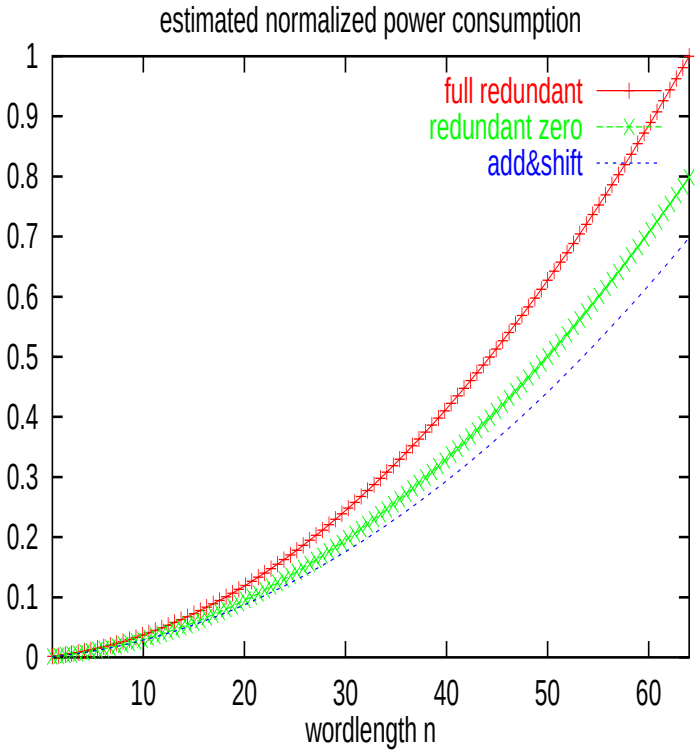


vectoring mode

Outline

- Motivation
- Introduction to CORDIC algorithm
- Previous approaches
- Area reduction method for add&shift algorithms
- Application to CORDIC
- Benefits of area reduction
- Conclusion

Benefits of area reduction



- power minimization
  - static power decreases proportional with cell area
  - dynamic power decreases as well
- speed improvment

  smaller chip area results in shorter wire length with reduced capacity load for standard cell layouts

Conclusion

- Area reduction method is applicable in general add&shift architectures by successively adder cells savings
- Up to 40% area savings possible
- Results checked by sample synthesized layouts
- With optimized full custom cells for CAB, RZ0, RZ the results can be improved

List of Slides

References

[Antelo et al., 1996] Antelo, E., Brugera, J., and Zapata, E. (1996). Unified mixed radix 2-4 redundant cordic processor. *IEEE Trans. on Computers*, 45(9):1068–1073.

[Lee and Lang, 1992] Lee, J.-A. and Lang, T. (1992). Constant-factor redundant cordic for angle calculation and rotation. *IEEE Trans. on Computers*, 41(8):1016–1025.

[Schmidt, et al., 1986] Schmidt, et al. (1986). Parameter optimization of the cordic-algorithm and implementation in a cmos-chip. In *Proc. EUSICO-86, B. 2*, pages 1219–1222, Hague, Netherlands.

[Takagi et al., 1991] Takagi, N., Asada, T., and Yajima, S. (1991). Redundant cordic methods with a constant scale factor for sine and cosine computation. *IEEE Trans. on Computers*, 40(9):989–995.

[Timmermann et al., 1992] Timmermann, D., Hahn, H., and Hosticka, B. (1992). Low latency time cordic algorithms. *IEEE Trans. on Computers*, 41(8):1010–1015.

[Timmermann and Sundsbø, 1992] Timmermann, D. and Sundsbø, I. (1992). Area and latency efficient cordic architectures. In *Proc. ISCAS'92*, pages 1093–1096, San Diego.