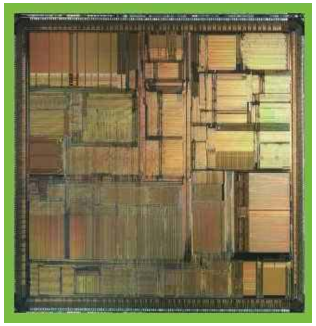
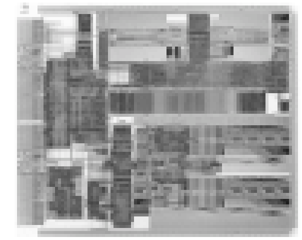
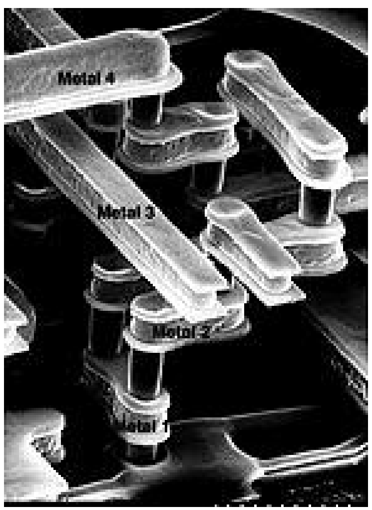




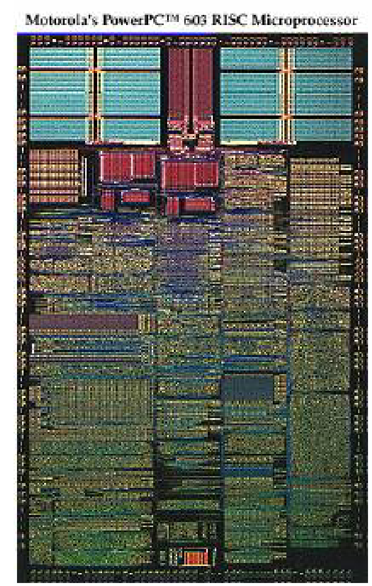
VLSI Arithmetic

Lecture 11:
Division

Prof. Vojin G. Oklobdzija
University of California



http://www.ece.ucdavis.edu/acsel



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Division

- Part Goals
  - Review shift-subtract division schemes
  - Learn about faster dividers
  - Discuss speed/cost tradeoffs in dividers
- Part Synopsis
  - Division is the hardest basic operation
  - Fortunately, it is also the least common
  - Division speedup: high-radix, array, ...
  - Combined multiplication/division hardware
  - Digit-recurrence vs convergence division

**from Parhami*

Shift/Subtract Division Algorithms

Notation for our discussion of division algorithms:

z	Dividend	$z_{2k-1}z_{2k-2} \cdots z_1z_0$
d	Divisor	$d_{k-1}d_{k-2} \cdots d_1d_0$
q	Quotient	$q_{k-1}q_{k-2} \cdots q_1q_0$
s	Remainder $(z - d \times q)$	$s_{k-1}s_{k-2} \cdots s_1s_0$

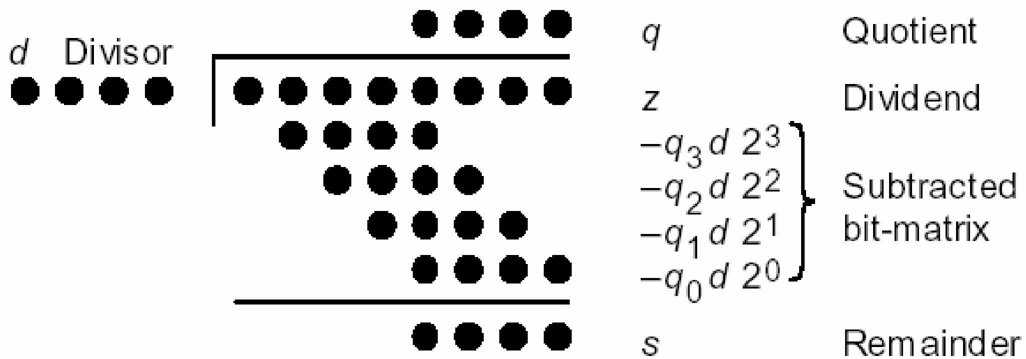


Fig. 13.1 Division of an 8-bit number by a 4-bit number in dot notation.

**from Parhami*

Division

- Division is more complex than multiplication:
  - Need for quotient digit selection or estimation
  - Possibility of overflow: the high-order *k* bits of *z*
  - must be strictly less than *d*; this overflow check
  - also detects the divide-by-zero condition.

**from Parhami*

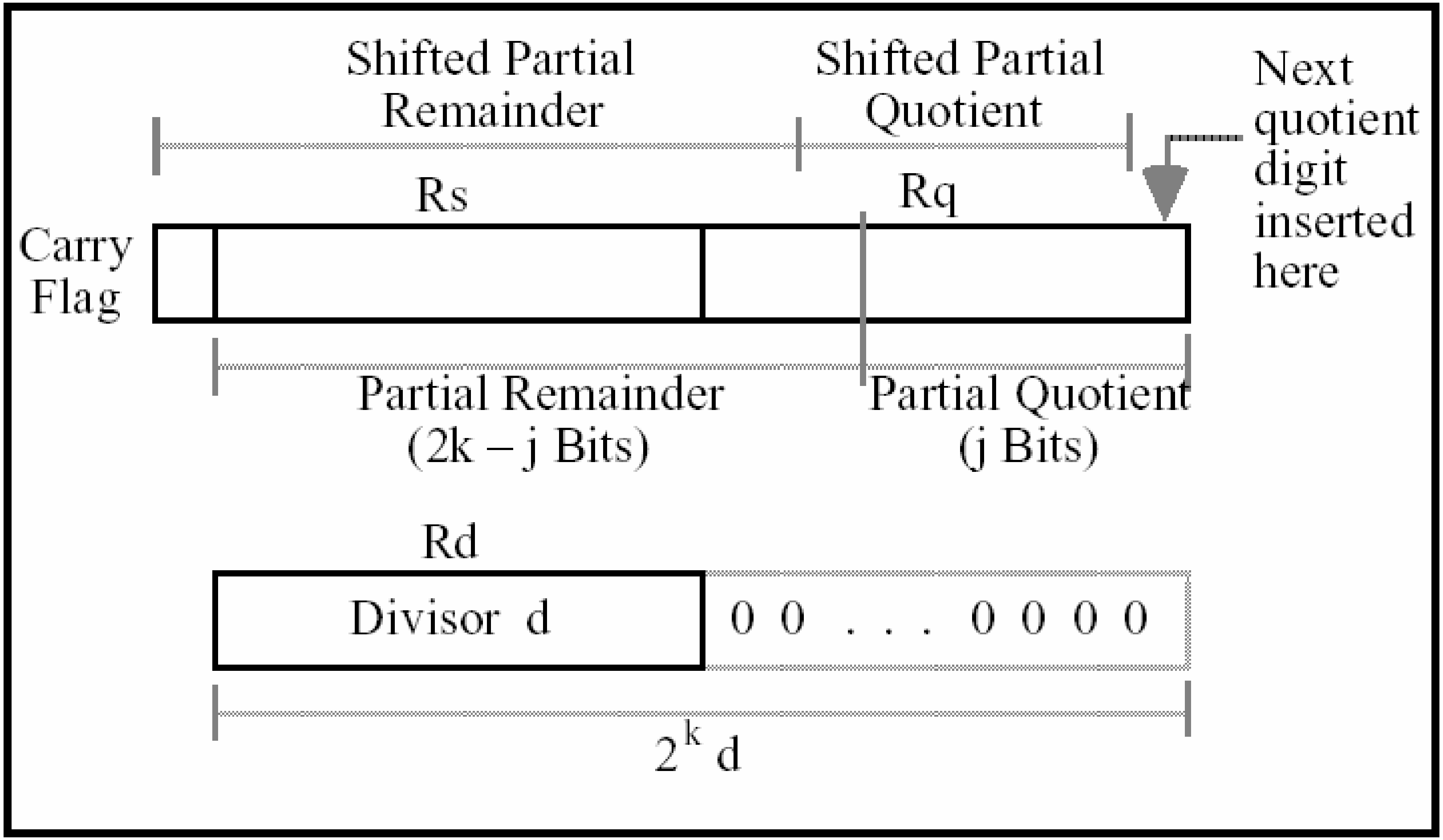
Division

Integer division		Fractional division	
z	0 1 1 1 0 1 0 1	z_{frac}	. 0 1 1 1 0 1 0 1
$2^4 d$	1 0 1 0	d_{frac}	. 1 0 1 0
$s^{(0)}$	0 1 1 1 0 1 0 1	$s^{(0)}$	. 0 1 1 1 0 1 0 1
$2s^{(0)}$	0 1 1 1 0 1 0 1	$2s^{(0)}$	0 . 1 1 1 0 1 0 1
$-q_3 2^4 d$	1 0 1 0 $\{q_3 = 1\}$	$-q_{-1} d$	. 1 0 1 0 $\{q_{-1} = 1\}$
$s^{(1)}$	0 1 0 0 1 0 1	$s^{(1)}$	. 0 1 0 0 1 0 1
$2s^{(1)}$	0 1 0 0 1 0 1	$2s^{(1)}$	0 . 1 0 0 1 0 1
$-q_2 2^4 d$	0 0 0 0 $\{q_2 = 0\}$	$-q_{-2} d$	. 0 0 0 0 $\{q_{-2} = 0\}$
$s^{(2)}$	1 0 0 1 0 1	$s^{(2)}$	. 1 0 0 1 0 1
$2s^{(2)}$	1 0 0 1 0 1	$2s^{(2)}$	1 . 0 0 1 0 1
$-q_1 2^4 d$	1 0 1 0 $\{q_1 = 1\}$	$-q_{-3} d$	. 1 0 1 0 $\{q_{-3} = 1\}$
$s^{(3)}$	1 0 0 0 1	$s^{(3)}$	. 1 0 0 0 1
$2s^{(3)}$	1 0 0 0 1	$2s^{(3)}$	1 . 0 0 0 1
$-q_0 2^4 d$	1 0 1 0 $\{q_0 = 1\}$	$-q_{-1} d$	. 1 0 1 0 $\{q_{-4} = 1\}$
$s^{(4)}$	0 1 1 1	$s^{(4)}$	. 0 1 1 1
s	0 1 1 1	s_{frac}	0 . 0 0 0 0 0 1 1 1
q	1 0 1 1	q_{frac}	. 1 0 1 1

Fig. 13.2 Examples of sequential division with integer and fractional operands.

**from Parhami*

Programmed Division



Register usage for programmed division.

**from Parhami*

Restoring Hardware Dividers

Division with signed operands: q and s are defined by

$$z = d \times q + s \qquad \text{sign}(s) = \text{sign}(z) \qquad |s| < |d|$$

Examples of division with signed operands

$$\begin{array}{llll}
z = 5 & d = 3 & \Rightarrow \quad q = 1 & s = 2 \\
z = 5 & d = -3 & \Rightarrow \quad q = -1 & s = 2 \\
z = -5 & d = 3 & \Rightarrow \quad q = -1 & s = -2 \\
z = -5 & d = -3 & \Rightarrow \quad q = 1 & s = -2
\end{array}$$

Magnitudes of q and s are unaffected by input signs

Signs of q and s are derivable from signs of z and d

Will discuss direct signed division later

**from Parhami*

Restoring Hardware Dividers

Division with signed operands: q and s are defined by

$$z = d \times q + s \qquad \text{sign}(s) = \text{sign}(z) \qquad |s| < |d|$$

Examples of division with signed operands

$$z = 5 \quad d = 3 \quad \Rightarrow \quad q = 1 \quad s = 2$$
$$z = 5 \quad d = -3 \quad \Rightarrow \quad q = -1 \quad s = 2$$
$$z = -5 \quad d = 3 \quad \Rightarrow \quad q = -1 \quad s = -2$$
$$z = -5 \quad d = -3 \quad \Rightarrow \quad q = 1 \quad s = -2$$

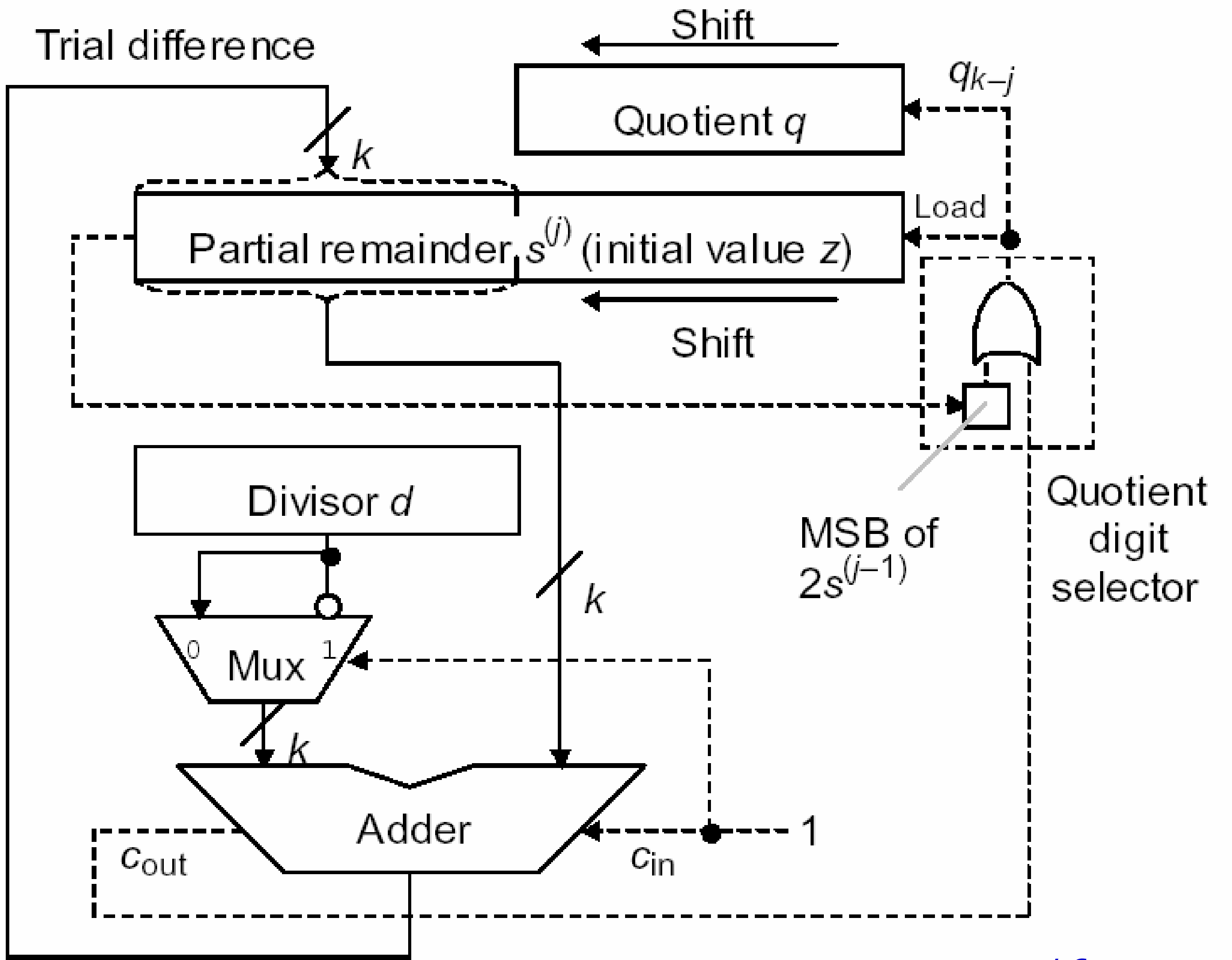
Magnitudes of q and s are unaffected by input signs

Signs of q and s are derivable from signs of z and d

Will discuss direct signed division later

**from Parhami*

Restoring Hardware Dividers



*from Parhami

Shift/subtract sequential restoring divider.

Restoring Division

z		0	1	1	1	0	1	0	1	No overflow, since: $(0111)_{two} < (1010)_{two}$
$2^4 d$	0	1	0	1	0					
$-2^4 d$	1	0	1	1	0					
$s^{(0)}$	0	0	1	1	1	0	1	0	1	
$2s^{(0)}$	0	1	1	1	0	1	0	1		
$+(-2^4 d)$	1	0	1	1	0					
$s^{(1)}$	0	0	1	0	0	1	0	1		Positive, so set $q_3 = 1$
$2s^{(1)}$	0	1	0	0	1	0	1			
$+(-2^4 d)$	1	0	1	1	0					
$s^{(2)}$	1	1	1	1	1	0	1			Negative, so set $q_2 = 0$ and restore
$s^{(2)} = 2s^{(1)}$	0	1	0	0	1	0	1			
$2s^{(2)}$	1	0	0	1	0	1				
$+(-2^4 d)$	1	0	1	1	0					
$s^{(3)}$	0	1	0	0	0	1				Positive, so set $q_1 = 1$
$2s^{(3)}$	1	0	0	0	1					
$+(-2^4 d)$	1	0	1	1	0					
$s^{(4)}$	0	0	1	1	1					Positive, so set $q_0 = 1$
s						0	1	1	1	
q						1	0	1	1	

**from Parhami*

Fig. 13.6 Example of restoring unsigned division.

Nonrestoring and Signed Division

The cycle time in restoring division must accommodate:

- shifting the registers
- allowing signals to propagate through the adder
- determining and storing the next quotient digit
- storing the trial difference, if required

Later events depend on earlier ones in the same cycle

Such dependencies tend to lengthen the clock cycle.

Nonrestoring division algorithm assume $q_{k-j} = 1$ and perform

- subtraction
- store the difference as the new partial remainder

(the partial remainder can become incorrect, hence the name "nonrestoring")

**from Parhami*

Nonrestoring Division

Why it is acceptable to store an incorrect value in the partial-remainder register?

Shifted partial remainder at start of the cycle is u
Subtraction yields the negative result $u - 2^k d$

Option 1: restore the partial remainder to correct value u,
shift, and subtract to get $2u - 2^k d$
Option 2: keep the incorrect partial remainder $u - 2^k d$,
shift, and add to get $2(u - 2^k d) + 2^k d = 2u - 2^k d$

**from Parhami*

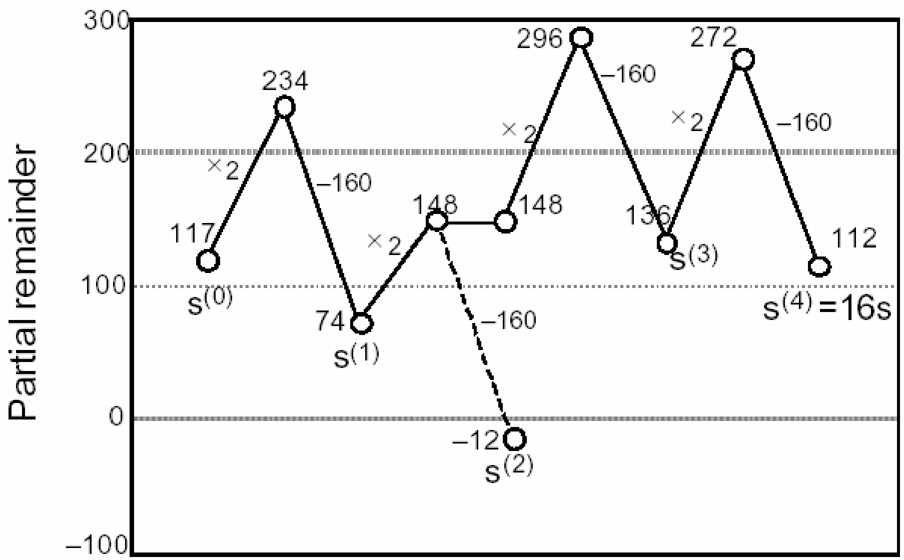
Non-restoring Division

z		0	1	1	1	0	1	0	1	No overflow, since:
$2^4 d$	0	1	0	1	0					$(0111)_{two} < (1010)_{two}$
$-2^4 d$	1	0	1	1	0					
$s^{(0)}$	0	0	1	1	1	0	1	0	1	
$2s^{(0)}$	0	1	1	1	0	1	0	1		Positive,
$+(-2^4 d)$	1	0	1	1	0					so subtract
$s^{(1)}$	0	0	1	0	0	1	0	1		
$2s^{(1)}$	0	1	0	0	1	0	1			Positive, so set $q_3 = 1$
$+(-2^4 d)$	1	0	1	1	0					and subtract
$s^{(2)}$	1	1	1	1	1	0	1			
$2s^{(2)}$	1	1	1	1	0	1				Negative, so set $q_2 = 0$
$+2^4 d$	0	1	0	1	0					and add
$s^{(3)}$	0	1	0	0	0	1				
$2s^{(3)}$	1	0	0	0	1					Positive, so set $q_1 = 1$
$+(-2^4 d)$	1	0	1	1	0					and subtract
$s^{(4)}$	0	0	1	1	1					Positive, so set $q_0 = 1$
s						0	1	1	1	
q						1	0	1	1	

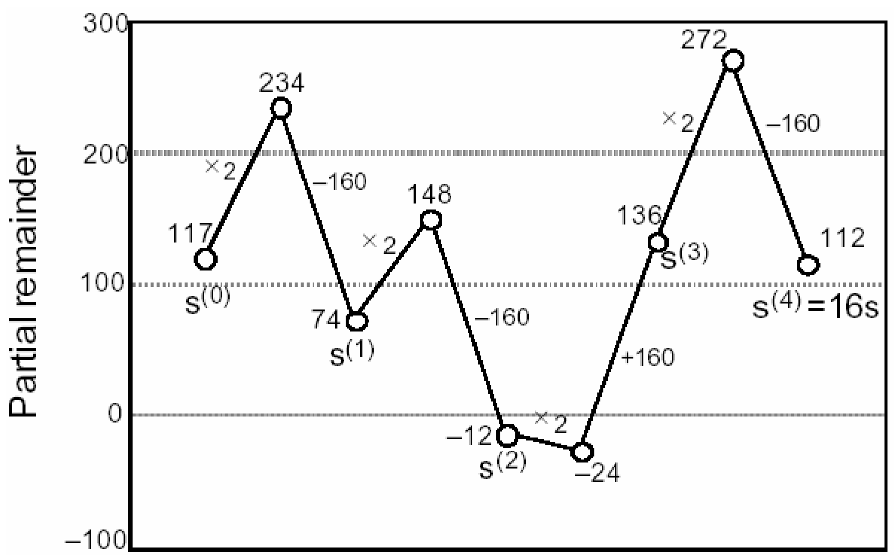
**from Parhami*

Fig. 13.7 Example of nonrestoring unsigned division.

Nonrestoring Division Example



(a) Restoring



(b) Nonrestoring

(0 1 1 1 0 1 0 1)two / (1 0 1 0)two

(117)ten / (10)ten

**from Parhami*

Division

Restoring division

$q_{k-j} = 0$ means no subtraction (or subtraction of 0)

$q_{k-j} = 1$ means subtraction of d

Nonrestoring division

We always subtract or add

As if quotient digits are selected from the set {1, $^-1$}

1 corresponds to subtraction

$^-1$ corresponds to addition

Our goal is to end up with a remainder that matches the sign of the dividend

This idea of trying to match the sign of s with the sign z, leads to a direct signed division algorithm

If sign(s) = sign(d) then $q_{k-j} = 1$ else $q_{k-j} = {}^-1$

*from Parhami

Division

Two problems must be dealt with at the end:

1. Converting the quotient with digits 1 and $^{-}1$ to binary
2. Adjusting the results if final remainder has wrong sign (correction step involves addition of $\pm d$ to remainder and subtraction of ± 1 from quotient)

Correction might be required even in unsigned division (when the final remainder is negative)

**from Parhami*

Division

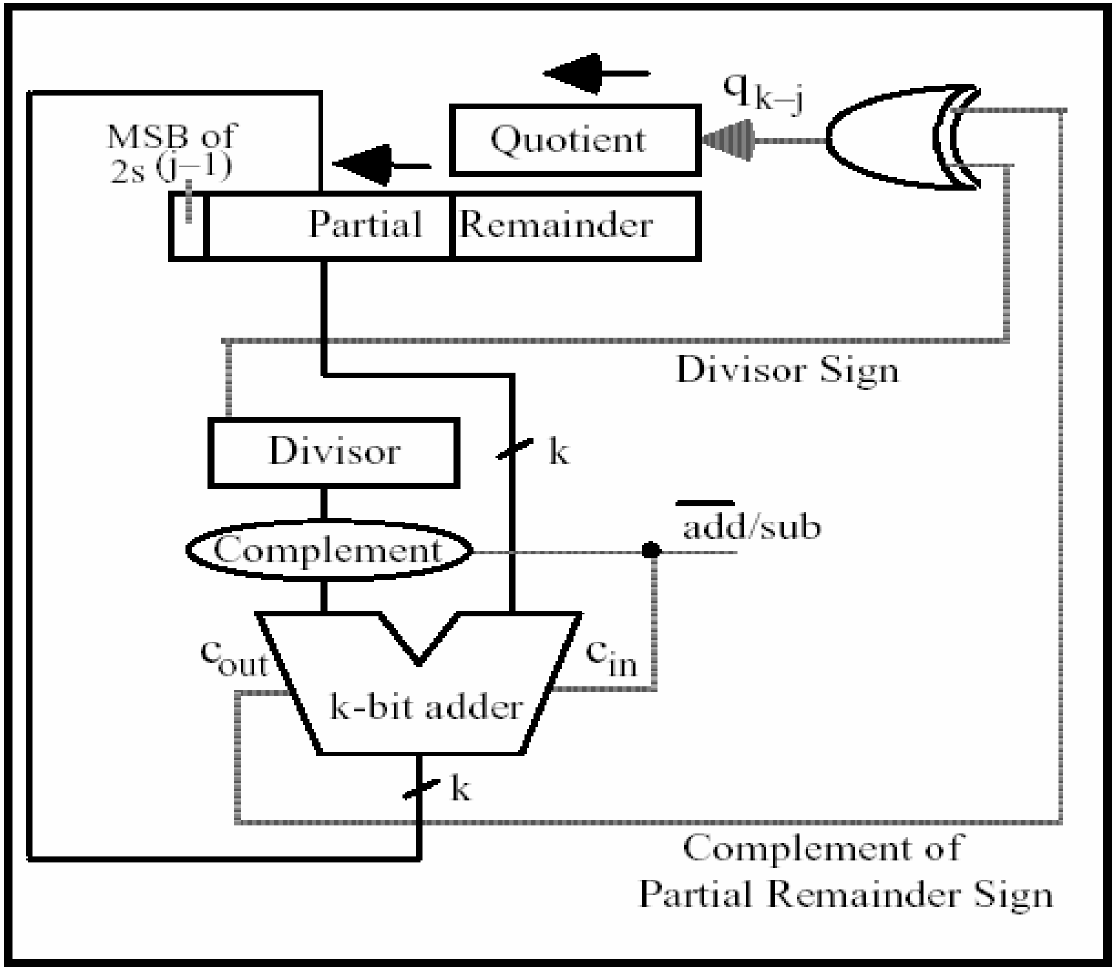
z		0	0	1	0	0	0	0	1	Dividend = $(33)_{ten}$
2^4d	1	1	0	0	1					Divisor = $(-7)_{ten}$
-2^4d	0	0	1	1	1					
$s^{(0)}$	0	0	0	1	0	0	0	0	1	
$2s^{(0)}$	0	0	1	0	0	0	0	1		sign($s^{(0)}$) • sign(d),
$+2^4d$	1	1	0	0	1					so set $q_3 = -1$ and add
$s^{(1)}$	1	1	1	0	1	0	0	1		
$2s^{(1)}$	1	1	0	1	0	0	1			sign($s^{(1)}$) = sign(d),
$+(-2^4d)$	0	0	1	1	1					so set $q_2 = 1$ and sub
$s^{(2)}$	0	0	0	0	1	0	1			
$2s^{(2)}$	0	0	0	1	0	1				sign($s^{(2)}$) • sign(d),
$+2^4d$	1	1	0	0	1					so set $q_1 = -1$ and add
$+(-2^4d)$	1	0	1	1	0					
$s^{(3)}$	1	1	0	1	1	1				
$2s^{(3)}$	1	0	1	1	1					sign($s^{(3)}$) = sign(d),
$+(-2^4d)$	0	0	1	1	1					so set $q_0 = 1$ and sub
$s^{(4)}$	1	1	1	1	0					sign($s^{(4)}$) • sign(z)
$+(-2^4d)$	0	0	1	1	1					Corrective subtraction
$s^{(4)}$	0	0	1	0	1					
s						0	1	0	1	Remainder = $(5)_{ten}$
q						-1	1	-1	1	Uncorrected BSD form
$q_{2's\text{-}compl}$						1	1	0	0	Corrected $q = (-4)_{ten}$

Fig. 13.9 Example of nonrestoring signed division.

**from Parhami*

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Division



Shift-subtract sequential nonrestoring divider.

**from Parhami*

14.1 Basics of High-Radix Division

Radix-r version of division recurrence of Section 13.1

$$s^{(j)} = r\,s^{(j-1)} - q_{k-j}(r^k d) \quad \text{with} \quad s^{(0)} = z \ \text{ and } \ s^{(k)} = r^k s$$

High-radix dividers of practical interest have $r = 2^b$ (and, occasionally, $r = 10$)

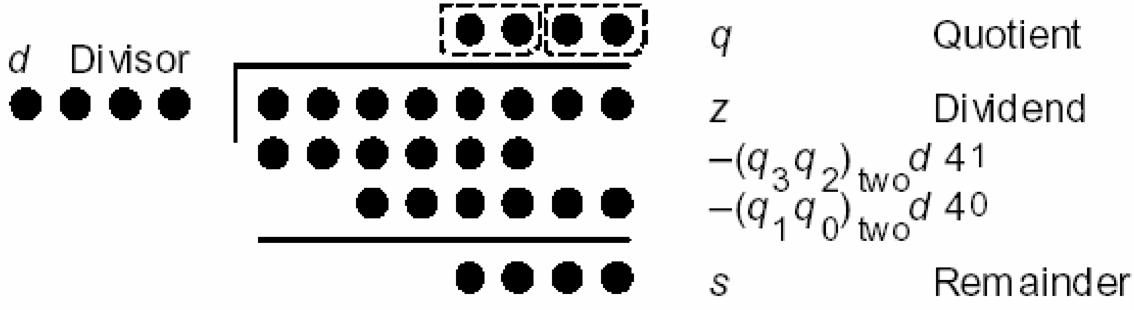


Fig. 14.1 Radix-4 division in dot notation.

**from Parhami*

High-Radix Division

Radix-4 integer division

z	0 1 2 3 1 1 2 3	
$4^4 d$	1 0 0 3	
$s^{(0)}$	0 1 2 3 1 1 2 3	
$4s^{(0)}$	0 1 2 3 1 1 2 3	
$-q_3\, 4^4 d$	0 1 2 0 3	$\{q_3 = 1\}$
$s^{(1)}$	0 0 2 2 1 2 3	
$4s^{(1)}$	0 0 2 2 1 2 3	
$-q_2\, 4^4 d$	0 0 0 0 0	$\{q_2 = 0\}$
$s^{(2)}$	0 2 2 1 2 3	
$4s^{(2)}$	0 2 2 1 2 3	
$-q_1\, 4^4 d$	0 1 2 0 3	$\{q_1 = 1\}$
$s^{(3)}$	1 0 0 3 3	
$4s^{(3)}$	1 0 0 3 3	
$-q_0\, 4^4 d$	0 3 0 1 2	$\{q_0 = 2\}$
$s^{(4)}$	1 0 2 1	
s	1 0 2 1	
q	1 0 1 2	

Radix-10 fractional division

z_{frac}	. 7 0 0 3	
d_{frac}	. 9 9	
$s^{(0)}$	. 7 0 0 3	
$10s^{(0)}$	7 . 0 0 3	
$-q_{-1} d$	6 . 9 3	$\{q_{-1} = 7\}$
$s^{(1)}$	. 0 7 3	
$10s^{(1)}$	0 . 7 3	
$-q_{-2} d$	0 . 0 0	$\{q_{-2} = 0\}$
$s^{(2)}$	. 7 3	
s_{frac}	. 0 0 7 3	
q_{frac}	. 7 0	

**from Parhami*

Fig. 14.2 Examples of high-radix division with integer and fractional operands.

High-Radix Division

Radix-2 nonrestoring division, fractional operands

$$s^{(j)} = 2s^{(j-1)} - q_{-j}\, d \quad \text{with} \quad s^{(0)} = z \quad \text{and} \quad s^{(k)} = 2^k s$$

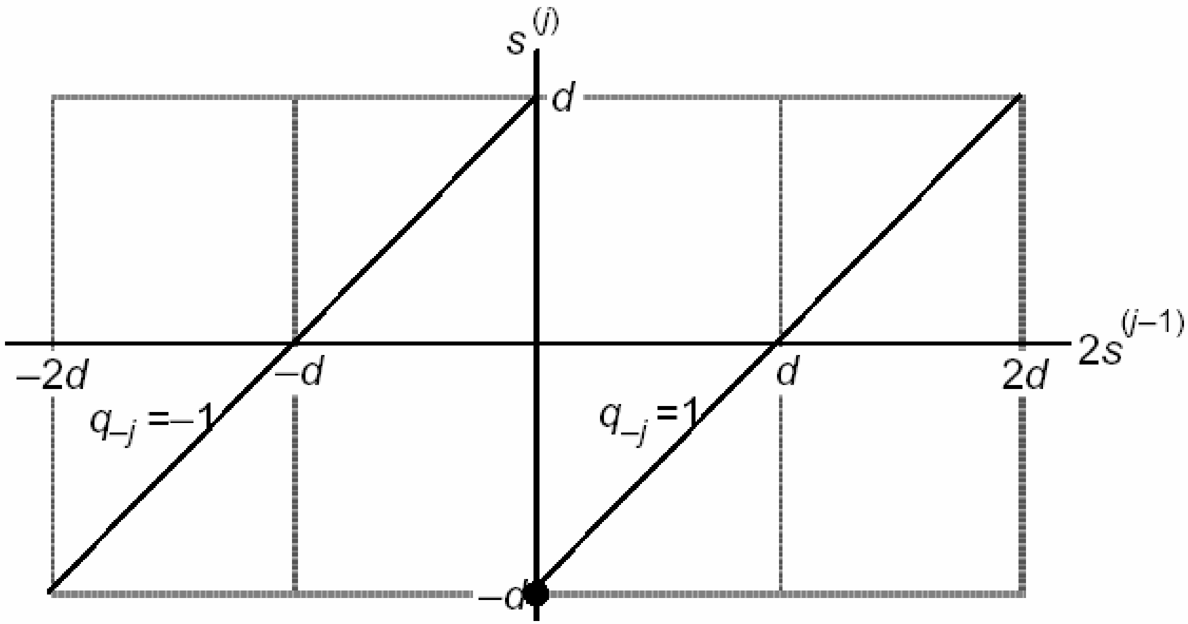


Fig. 14.3 The new partial remainder, $s^{(j)}$, as a function of the shifted old partial remainder, $2s^{(j-1)}$, in radix-2 nonrestoring division.

**from Parhami*

High-Radix Division

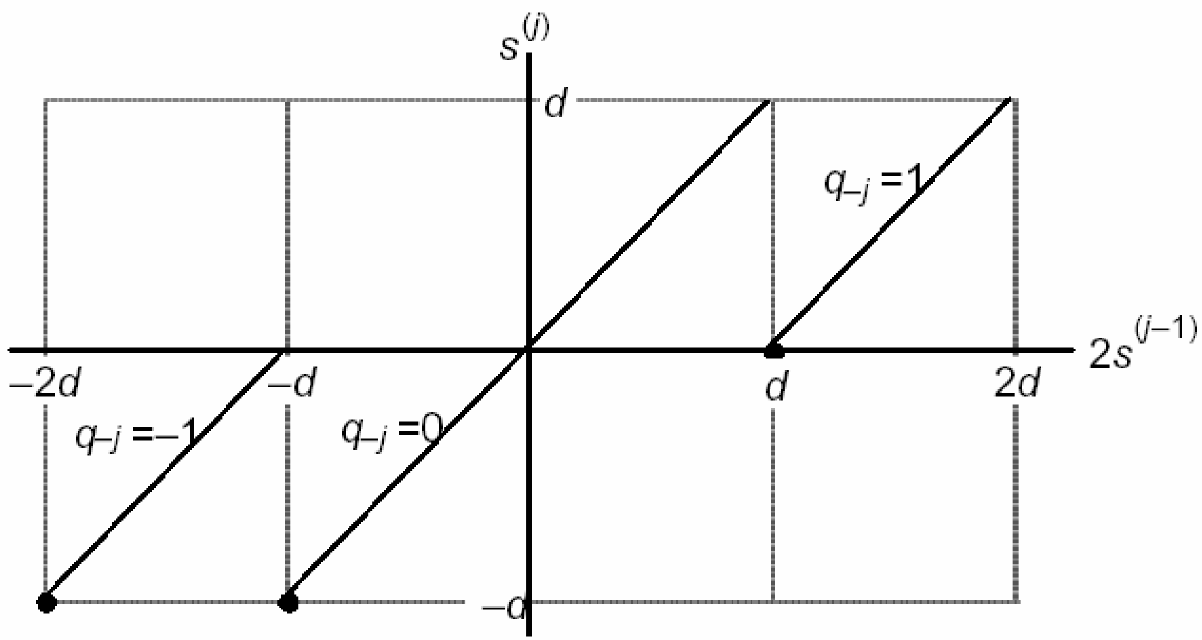


Fig. 14.4 The new partial remainder $s^{(j)}$ as a function of $2s^{(j-1)}$, with q_{-j} in {–1, 0, 1}.

**from Parhami*

SRT Division

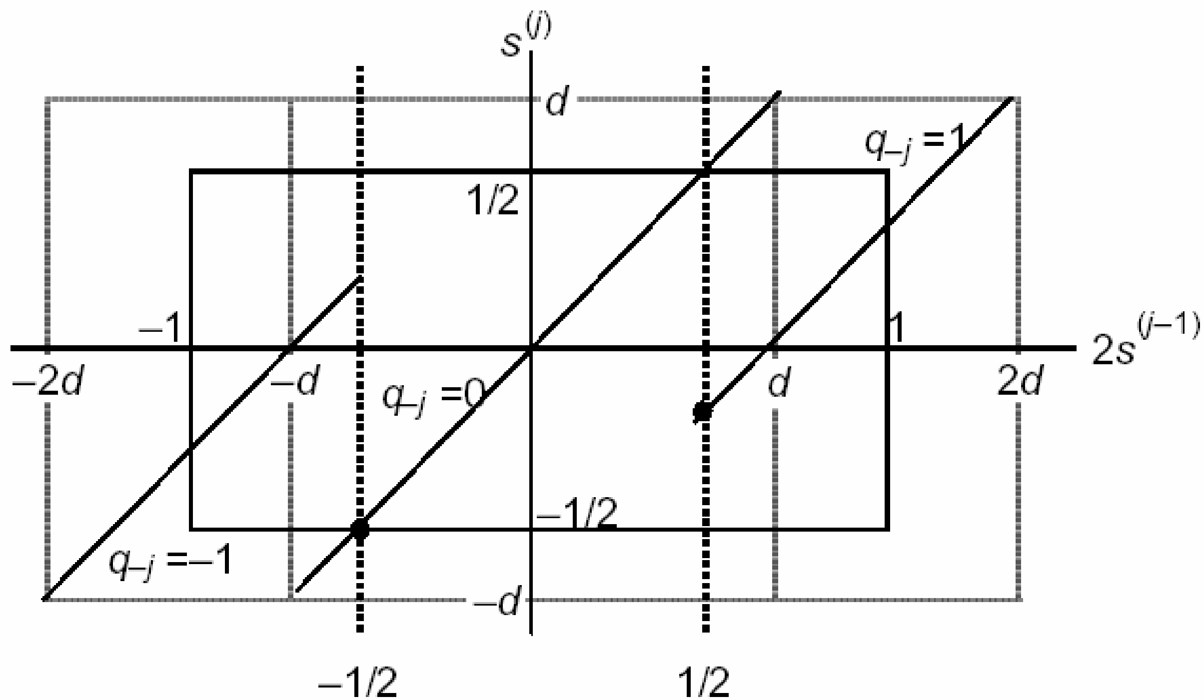


Fig. 14.5 The relationship between new and old partial remainders in radix-2 SRT division.

**from Parhami*

SRT Division

SRT algorithm (Sweeney, Robertson, Tocher)

$2s^{(j-1)} \geq +1/2 = (0.1)_{2\text{'s-compl}}$

$$\Rightarrow \quad 2s^{(j-1)} = (0.1u_{-2}u_{-3}\cdots)_{2\text{'s-compl}}$$

$2s^{(j-1)} < -1/2 = (1.1)_{2\text{'s-compl}}$

$$\Rightarrow \quad 2s^{(j-1)} = (1.0u_{-2}u_{-3}\cdots)_{2\text{'s-compl}}$$

Skipping over identical leading bits by shifting

$s^{(j-1)} = 0.0000110\ldots$ Shift left by 4 bits and subtract; append q with 0 0 0 1

$s^{(j-1)} = 1.1110100\ldots$ Shift left by 3 bits and add; append q with 0 0 $\bar{1}$

Average skipping distance (statistically): 2.67 bits

**from Parhami*

SRT Division

z	. 0 1 0 0 0 1 0 1	In [–1/2,1/2), so OK
d	. 1 0 1 0	In [1/2,1), so OK
$-d$	1 . 0 1 1 0	
$s^{(0)}$	0 . 0 1 0 0 0 1 0 1	
$2s^{(0)}$	0 . 1 0 0 0 1 0 1	$\geq 1/2$, so set $q_{-1} = 1$
$+(-d)$	1 . 0 1 1 0	and subtract
$s^{(1)}$	1 . 1 1 1 0 1 0 1	
$2s^{(1)}$	1 . 1 1 0 1 0 1	In [–1/2,1/2), so $q_{-2} = 0$
$s^{(2)} = 2s^{(1)}$	1 . 1 1 0 1 0 1	
$2s^{(2)}$	1 . 1 0 1 0 1	In [–1/2, 0), so $q_{-3} = 0$
$s^{(3)} = 2s^{(2)}$	1 . 1 0 1 0 1	
$2s^{(3)}$	1 . 0 1 0 1	< –1/2, so $q_{-4} = -1$
$+d$	0 . 1 0 1 0	and add
$s^{(4)}$	1 . 1 1 1 1	Negative,
$+d$	0 . 1 0 1 0	so add to correct
$s^{(4)}$	0 . 1 0 0 1	
s	0 . 0 0 0 0 1 0 0 1	
q	0 . 1 0 0 -1	Ucorrected BSD form
q	0 . 0 1 1 0	Convert, subtract *ulp*

**from Parhami*

Fig. 14.6 Example of unsigned radix-2 SRT division.

SRT Division

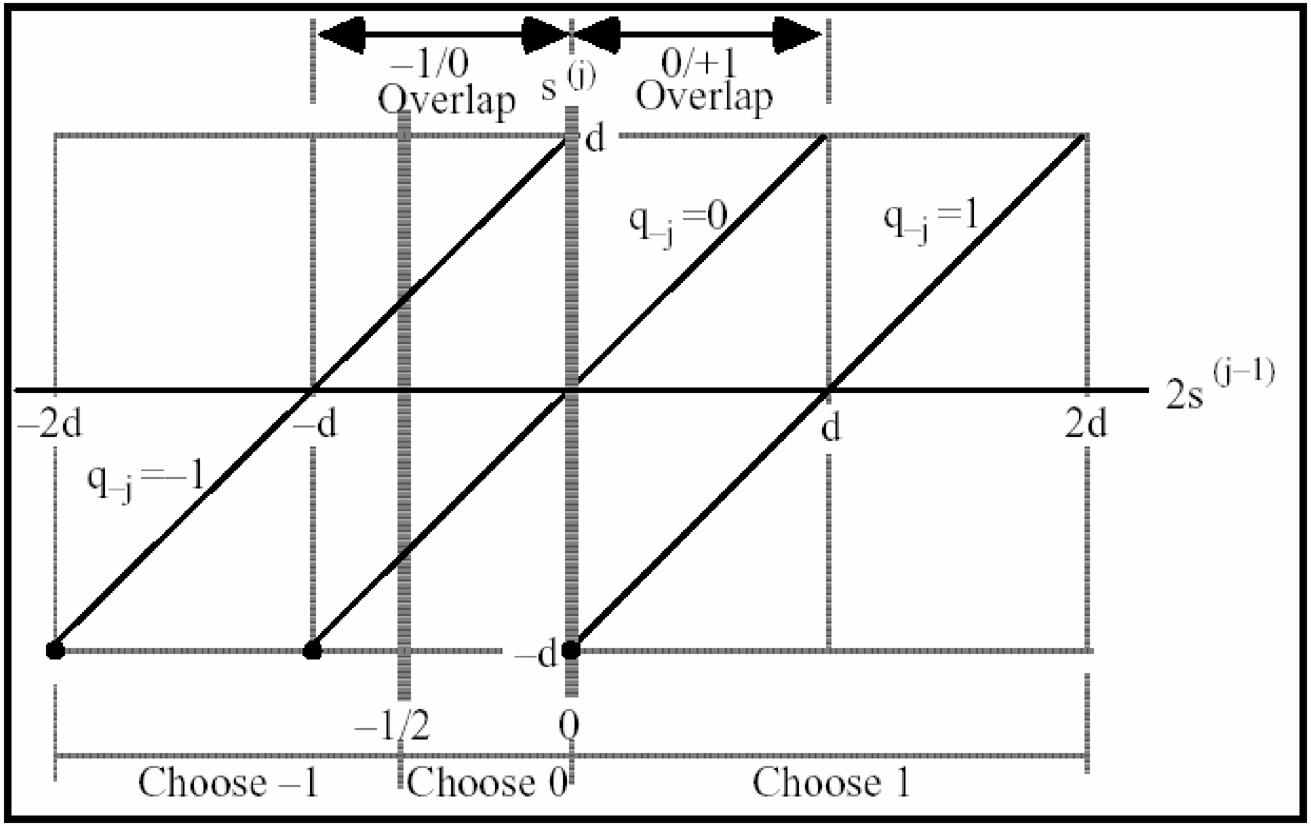


Fig. 14.7 Constant thresholds used for quotient digit selection in radix-2 division with q_{k-j} in {–1, 0, 1}.

**from Parhami*

Using Carry-Save Adder

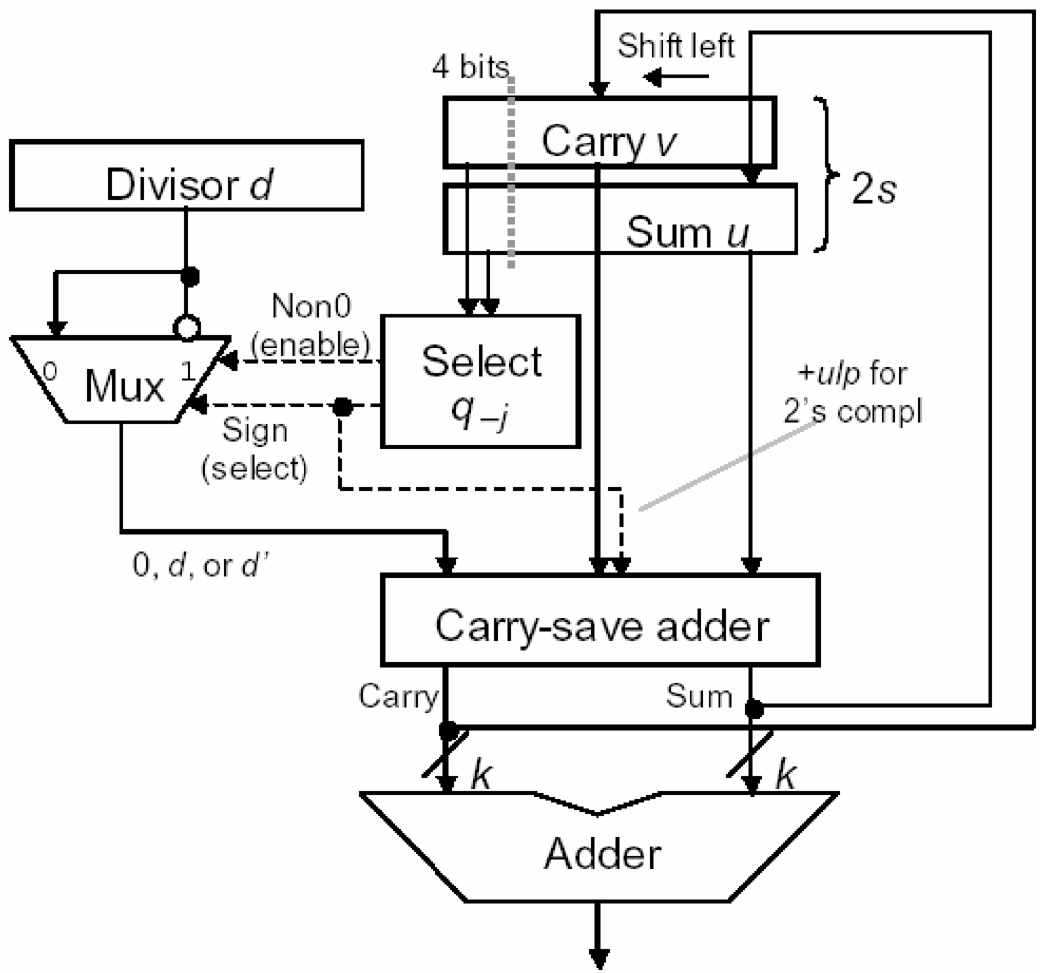
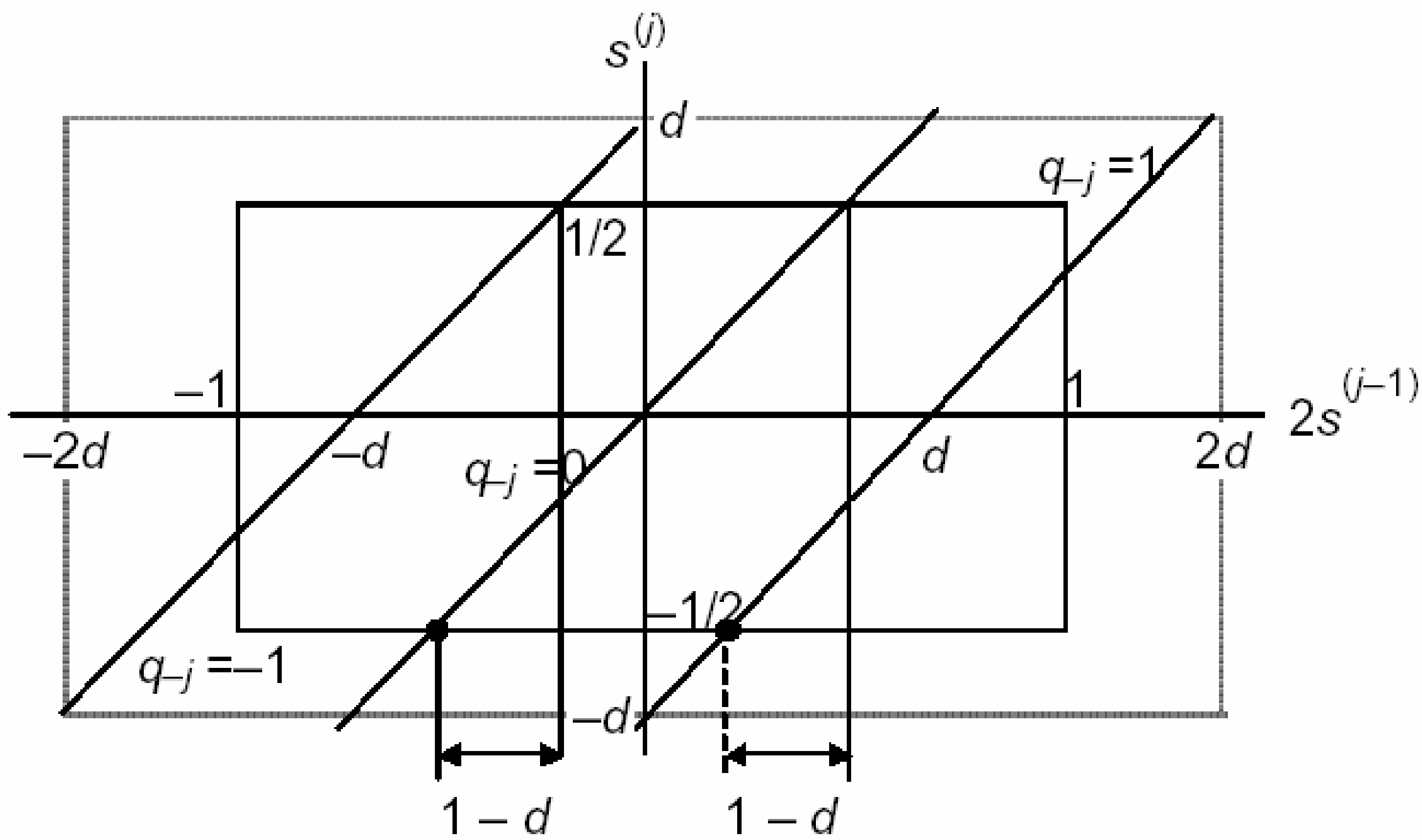


Fig. 14.8 Block diagram of a radix-2 divider with partial remainder in stored-carry form.

**from Parhami*

SRT Division



Overlap regions in radix-2 SRT division.

*from Parhami

14.4 Choosing the Quotient Digits

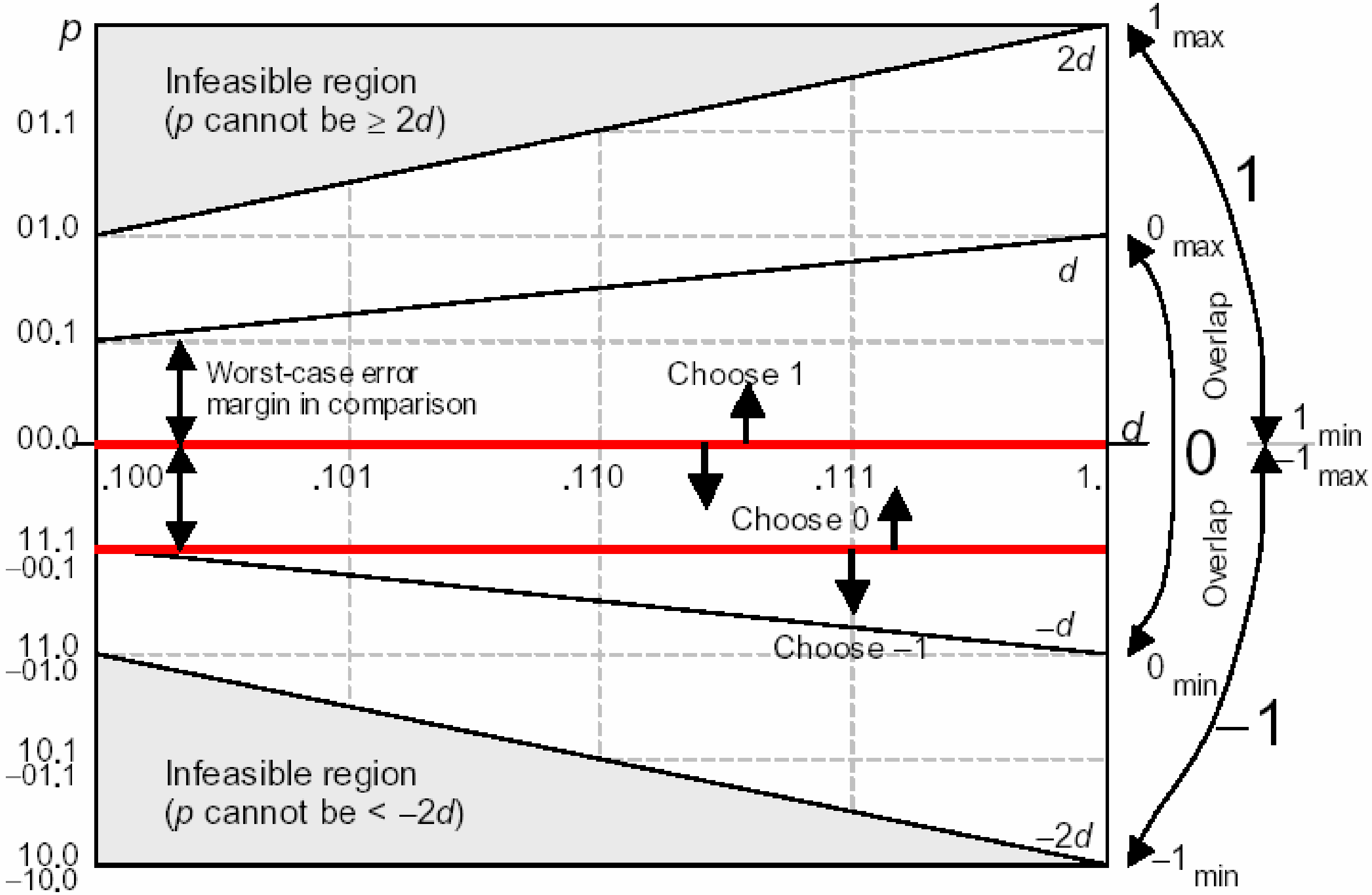


Fig. 14.10 **A *p*-*d* plot for radix-2 division with $d \in [1/2,1)$, partial remainder in $[-d, d)$, and quotient digits in $[-1, 1]$.**

**from Parhami*

Radix-4 SRT Division

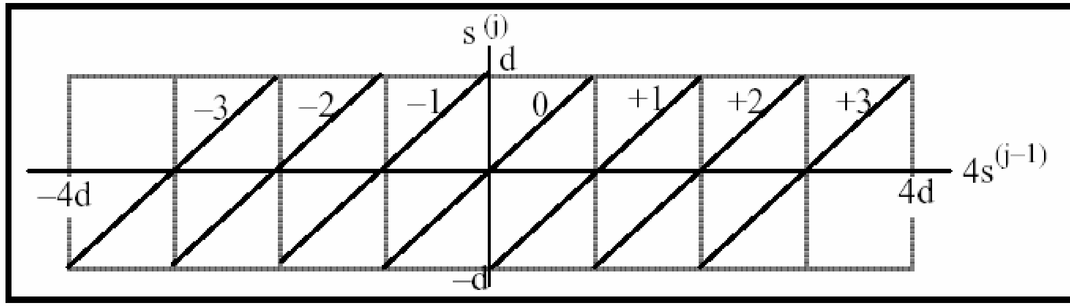


Fig. 14.11 New versus shifted old partial remainder in radix-4 division with q_{-j} in [–3, 3].

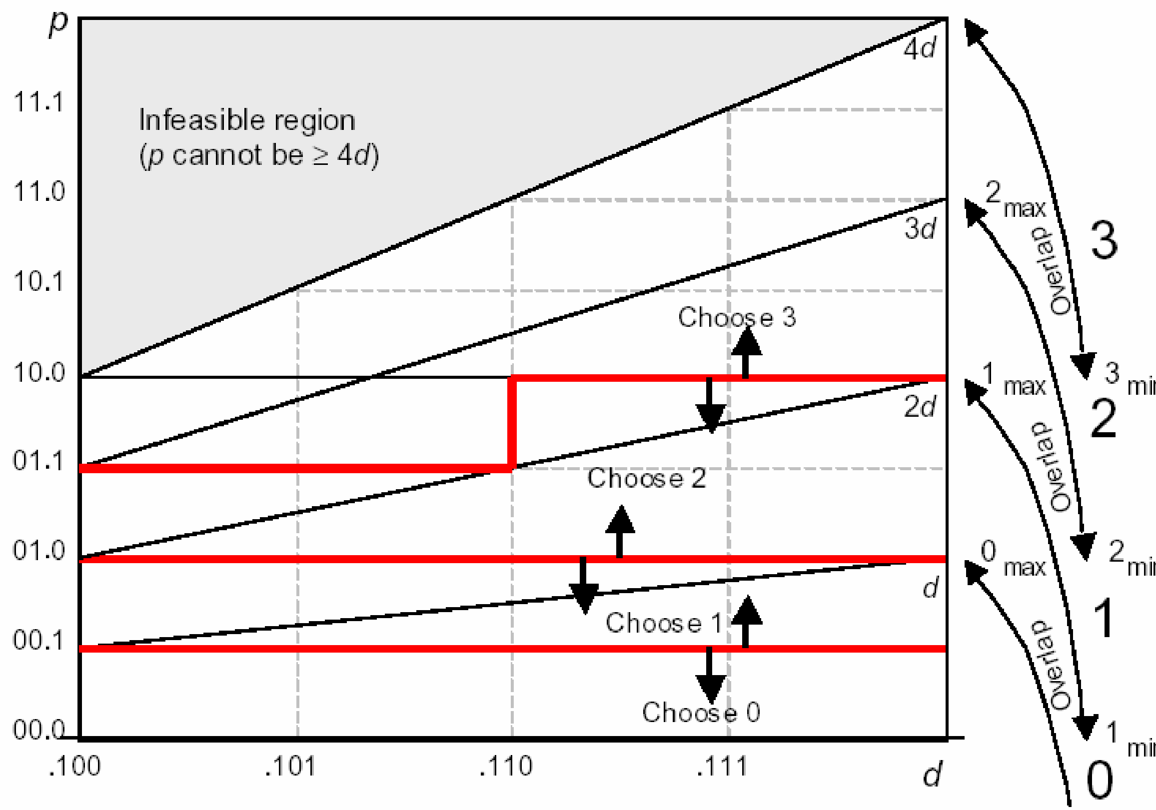


Fig. 14.12 *p-d* plot for radix-4 SRT division with quotient digit set [–3, 3].

**from Parhami*

General High-Radix Dividers

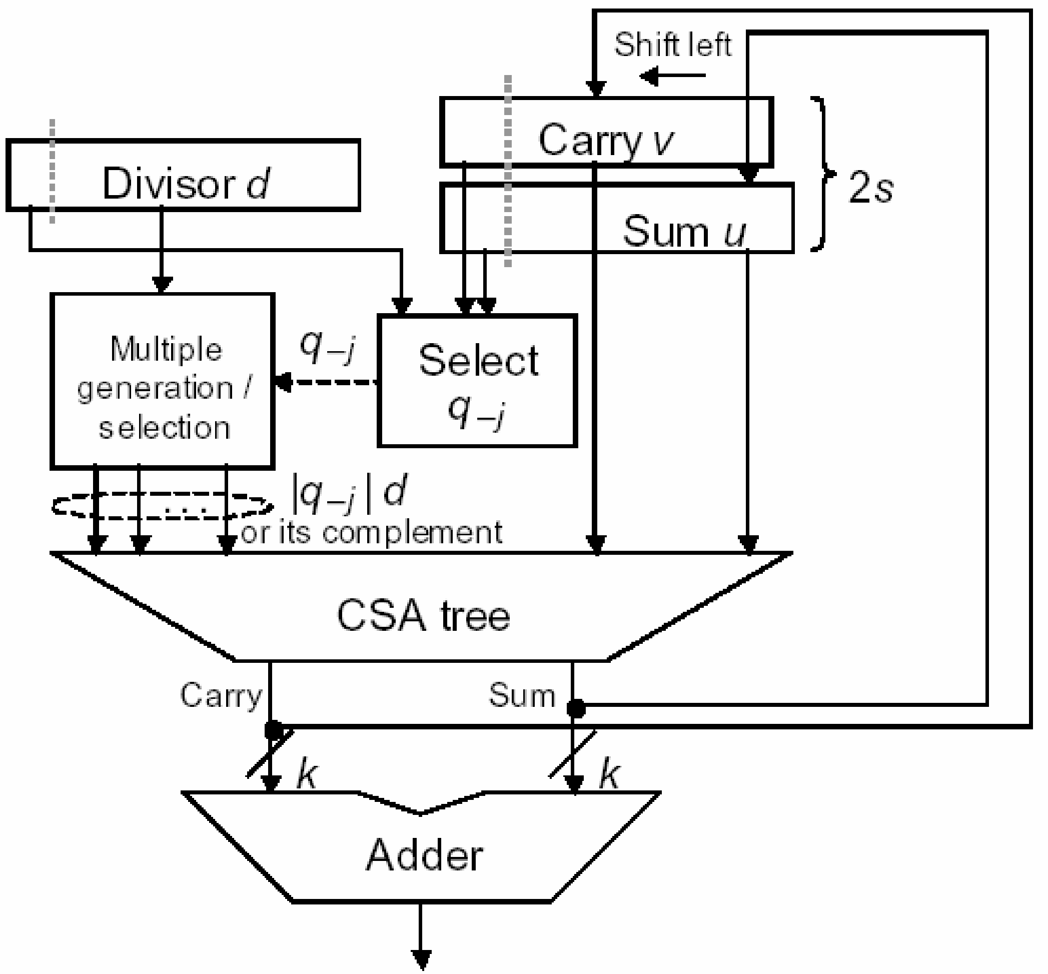
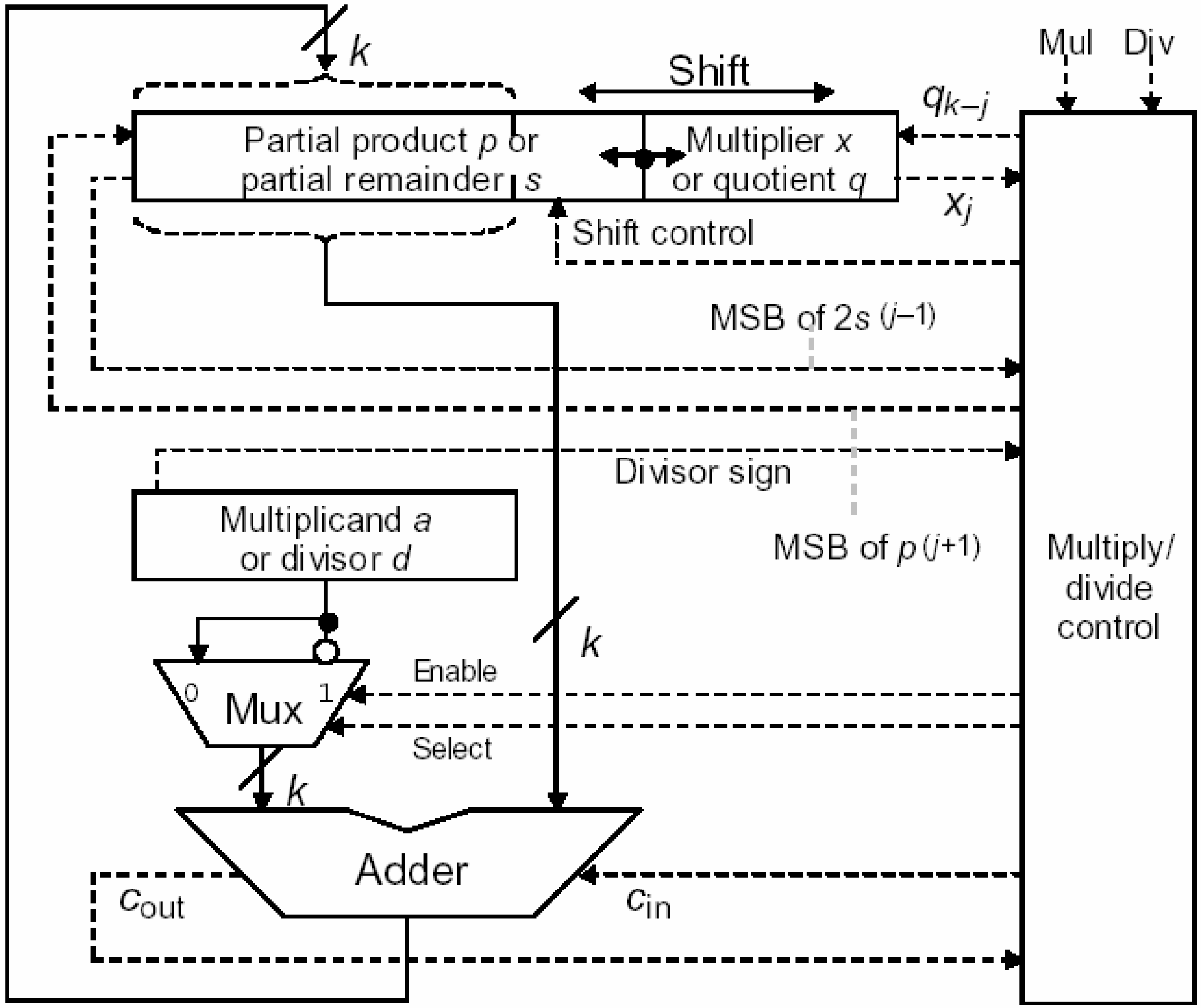


Fig. 14.15 Block diagram of radix-*r* divider with partial remainder in stored-carry form.

**from Parhami*

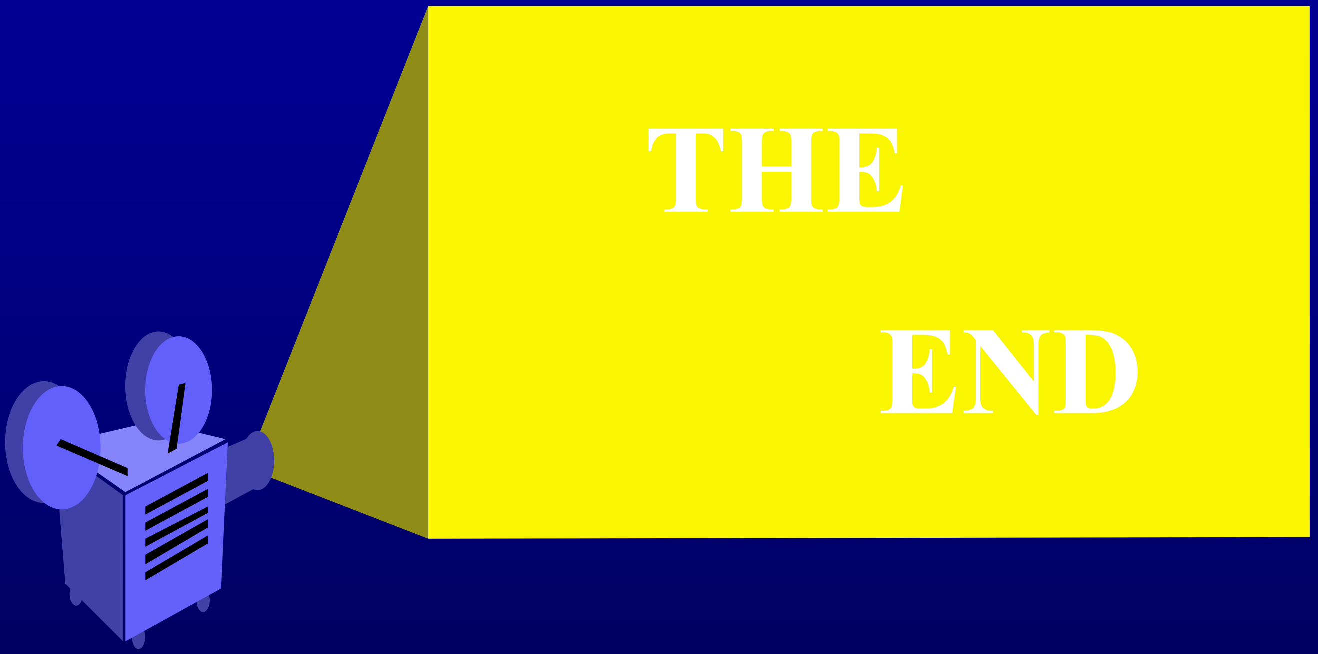
Combined Multiply/Divide Units

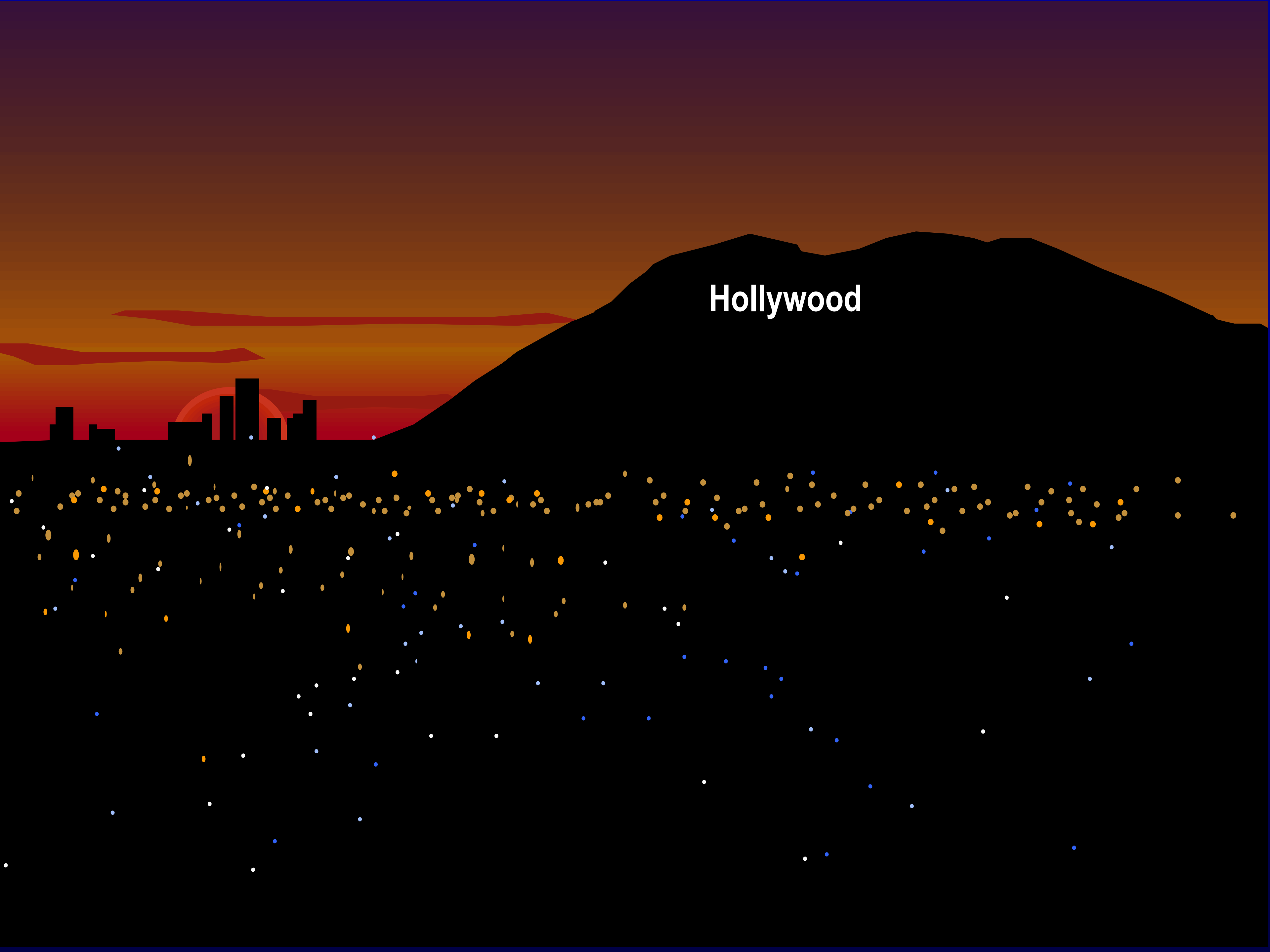


Sequential radix-2 multiply/divide unit.

Other Methods for Division

- General High-Radix Dividers
- Division with prescaling
- Array Dividers
- Division by Convergence
  - Division by repeated multiplication
  - Division by reciprocation


THE
END


Hollywood