UMASS
UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Digital Computer Arithmetic
ECE 666

Part 7
Fast Division

Israel Koren

Spring 2004

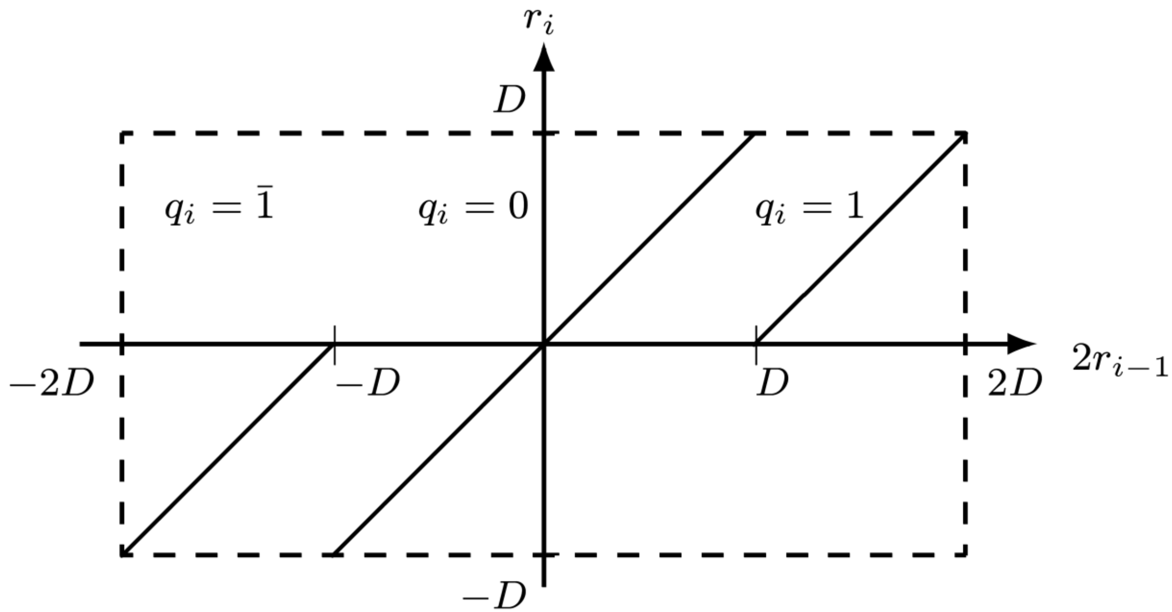
Fast Division - SRT Algorithm

- **2 approaches:**
  - First - conventional - uses add/subtract+shift, number of operations linearly proportional to word size n
  - Second - uses multiplication, number of operations logarithmic in n, but each step more complex
  - SRT - first approach
- Most well known division algorithm - named after Sweeney, Robertson, and Tocher
- Speed up nonrestoring division (n add/subtracts) - allows 0 as a quotient digit - no add/subtract:

$$q_i = \begin{cases} 1 & \text{if} \quad 2r_{i-1} \geq D \\ 0 & \text{if} \quad -D \leq 2r_{i-1} < D \\ \bar{1} & \text{if} \quad 2r_{i-1} < -D \end{cases}$$

$$r_i = 2r_{i-1} - q_i \cdot D$$

Modified Nonrestoring Division



- **Problem**: full comparison of $2r_{i-1}$ with either D or $-D$ required
- **Solution**: restricting D to normalized fraction $1/2 \leq |D| < 1$
- Region of $2r_{i-1}$ for which $q_i=0$ reduced to

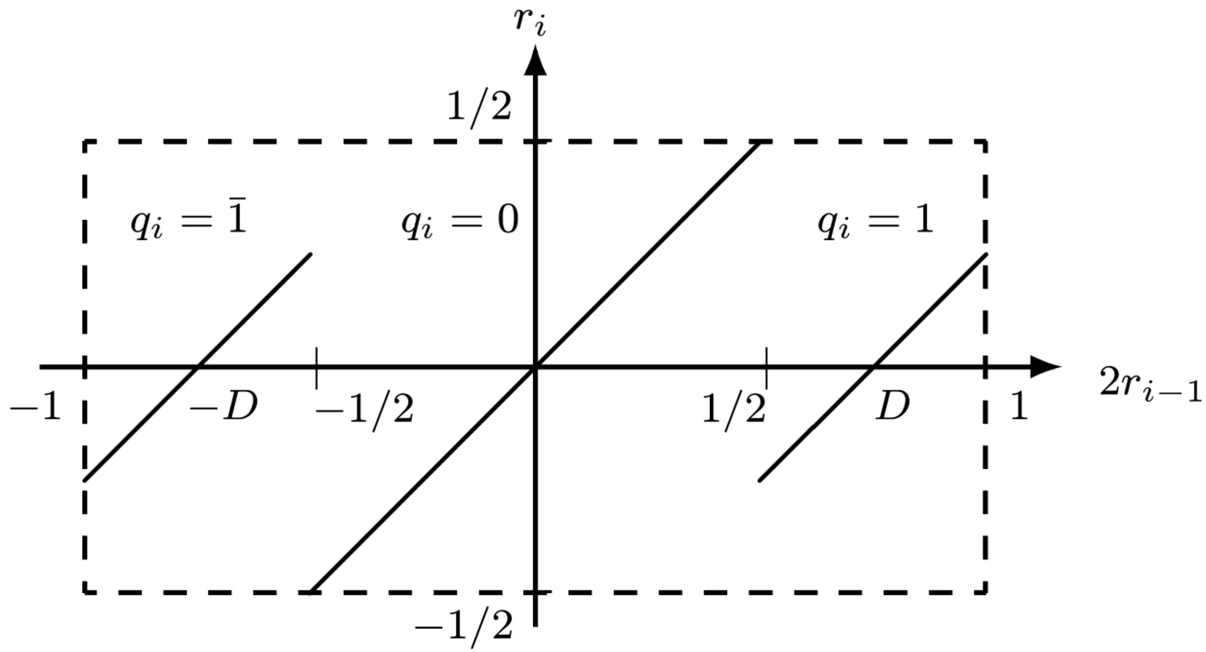
$$-D \leq -\frac{1}{2} \leq 2r_{i-1} < \frac{1}{2} \leq D$$

Modified Nonrestoring $\rightarrow$ SRT

- **Advantage:** Comparing partial remainder $2r_{i-1}$ to $1/2$ or $-1/2$, not D or $-D$
- Binary fraction in two's complement representation
  * $\geq 1/2$ if and only if it starts with 0.1
  * $\leq -1/2$ if and only if it starts with 1.0
- Only 2 bits of $2r_{i-1}$ examined - not full comparison between $2r_{i-1}$ and D
  * In some cases (e.g., dividend $X>1/2$) - shifted partial remainder needs an integer bit in addition to sign bit - 3 bits of $2r_{i-1}$ examined
- Selecting quotient digit:

$$q_i = \begin{cases} 1 & \text{if } \; 2r_{i-1} \geq 1/2 \\ 0 & \text{if } \; -1/2 \leq 2r_{i-1} < 1/2 \\ \bar{1} & \text{if } \; 2r_{i-1} < -1/2. \end{cases}$$

SRT Division Algorithm



- Quotient digits selected so $|r_i| \leq |D| \Rightarrow$ final remainder $< |D|$
- Process starts with normalized divisor - normalizing partial remainder by shifting over leading 0's/1's if positive/negative
- Example:
  - * $2r_{i-1}$=0.001xxxx (x - 0/1); $2r_{i-1}<1/2$ - set q_i=0, $2r_i$=0.01xxxx and so on
  - * $2r_{i-1}$=1.110xxxx; $2r_{i-1}>-1/2$ - set q_i=0, $2r_i$=1.10xxxx
- SRT is nonrestoring division with normalized divisor and remainder

Extension to Negative Divisors

$$q_i = \begin{cases} 0 & \text{if } |2r_{i-1}| < 1/2 \\ 1 & \text{if } |2r_{i-1}| \geq 1/2 \ \& \ r_{i-1} \text{ and } D \text{ have the same sign} \\ \bar{1} & \text{if } |2r_{i-1}| \geq 1/2 \ \& \ r_{i-1} \text{ and } D \text{ have opposite signs} \end{cases}$$

- **Example:** Dividend $X=(0.0101)_2 = 5/16$ Divisor $D=(0.1100)_2 = 3/4$

$r_0 = X$		0	.0	1	0	1	
$2r_0$		0	.1	0	1	0	$\geq 1/2$ set $q_1 = 1$
Add $-D$	$+$	1	.0	1	0	0	
r_1		1	.1	1	1	0	
$2r_1 = r_2$		1	.1	1	0	0	$\geq -1/2$ set $q_2 = 0$
$2r_2 = r_3$		1	.1	0	0	0	$\geq -1/2$ set $q_3 = 0$
$2r_3$		1	.0	0	0	0	$< -1/2$ set $q_4 = \bar{1}$
Add D	$+$	0	.1	1	0	0	
r_4		1	.1	1	0	0	negative remainder & positive X
Add D	$+$	0	.1	1	0	0	correction
r_4		0	.1	0	0	0	corrected final remainder

- **Before correction** $Q=0.100\bar{1}$ - minimal SD repr. of $Q=0.0111$ - minimal number of add/subtracts
- **After correction,** $Q = 0.0111 - \text{ulp} = 0.0110_2 = 3/8$; final remainder = $1/2 \cdot 2^{-4} = 1/32$

Example

- $X=(0.00111111)_2=63/256$ $D=(0.1001)_2=9/16$

$r_0 = X$		0	.0	0	1	1	1	1	1	1	
$2r_0$		0	.0	1	1	1	1	1	1	0	$< 1/2$ set $q_1 = 0$
$2r_1$		0	.1	1	1	1	1	1	0	0	$\geq 1/2$ set $q_2 = 1$
Add $-D$	$+$	1	.0	1	1	1					
r_2		0	.0	1	1	0	1	1	0	0	
$2r_2$		0	.1	1	0	1	1	0	0	0	$\geq 1/2$ set $q_3 = 1$
Add $-D$	$+$	1	.0	1	1	1					
r_3		0	.0	1	0	0	1	0	0	0	
$2r_3$		0	.1	0	0	1	0	0	0	0	$\geq 1/2$ set $q_4 = 1$
Add $-D$	$+$	1	.0	1	1	1					
r_4		0	.0	0	0	0	0	0	0	0	zero final remainder

- $Q=0.0111_2=7/16$ - **not a minimal representation in SD form**
- **Conclusion: Number of add/subtracts can be reduced further**

Properties of SRT

- Based on simulations and analysis:
- 1.Average "shift"=2.67 - $n/2.67$ operations for dividend of length n
  - $24/2.67 \sim 9$ operations on average for $n=24$
- 2.Actual number of operations depends on divisor D - smallest when $17/28 \leq D \leq 3/4$ - average shift of 3
- If D out of range ($3/5 \leq D \leq 3/4$) - SRT can be modified to reduce number of add/subtracts
- 2 ways to modify SRT

Two Modifications of SRT

- **Scheme 1:** In some steps during division -
  - If D too small - use a multiple of D like 2D
  - If D too large - use D/2
  - Subtracting 2D (D/2) instead of D - equivalent to performing subtraction one position earlier (later)
- **Motivation for Scheme 1:**
  - Small D may generate a sequence of 1's in quotient one bit at a time, with subtract operation per each bit
  - Subtracting 2D instead of D (equivalent to subtracting D in previous step) may generate negative partial remainder, generating sequence of 0's as quotient bits while normalizing partial remainder
- **Scheme 2:** Change comparison constant K=1/2 if D outside optimal range - allowed because ratio D/K matters - partial remainder compared to K not D

Example - Scheme 1 (Using 2D)

- Same as previous example -
- $X=(0.00111111)_2=63/256$ $D=(0.1001)_2=9/16$

$r_0 = X$		0	.0	0	1	1	1	1	1	1	
$2r_0$		0	.0	1	1	1	1	1	1	0	$< 1/2$ set $q_1 = 0$
$2r_1$	0	0	.1	1	1	1	1	1	0	0	subtract $2D$
Add $-2D$ +	1	0	.1	1	1						instead of D
r_2	1	1	.1	1	0	1	1	1	0	0	set $q_1 = 1$ and $q_2 = 0$
$2r_2$		1	.1	0	1	1	1	0	0	0	set $q_3 = 0$
$2r_3$		1	.0	1	1	1	0	0	0	0	$\leq -1/2$ set $q_4 = \bar{1}$
Add D +		0	.1	0	0	1					
r_4		0	.0	0	0	0	0	0	0	0	zero final remainder

- $Q = 0.100\bar{1}_2 = 7/16$ - minimal SD representation

Scheme 1 (Using D/2)

- Large D - one 0 in sequence of 1's in quotient may result in 2 consecutive add/subtracts instead of one
- Adding D/2 instead of D for last 1 before the single 0 - equivalent to performing addition one position later - may generate negative partial remainder
- Allows properly handling single 0
- Then continue normalizing partial remainder until end of sequence of 1's

Example

- $X=(0.01100)_2=3/8$; $D=(0.11101)_2=29/32$
- Correct 5-bit quotient - $Q=(0.01101)_2=13/32$
- Applying basic SRT algorithm - $Q=0.10\bar{1}1\bar{1}$ - single 0 not handled efficiently
- Using multiple D/2 -

$r_0 = X$		0	.0	1	1	0	0		
$2r_0$		0	.1	1	0	0	0		$\geq 1/2$ set $q_1 = 1$
Add $-D$	$+$	1	.0	0	0	1	1		
r_1		1	.1	1	0	1	1		
$2r_1$		1	.1	0	1	1	0		set $q_2 = 0$
$2r_2$		1	.0	1	1	0	0	0	add $D/2$ $q_3 = \bar{1})$
Add $D/2$	$+$	0	.0	1	1	1	0	1	instead of D
r_3		1	.1	1	0	1	0	1	set $q_3 = 0$ and
$2r_3$		1	.1	0	1	0	1		$q_4 = \bar{1}$
$2r_4$		1	.0	1	0	1	0		$\leq -1/2$ set $q_5 = \bar{1}$
Add D	$+$	0	.1	1	1	0	1		
r_5		0	.0	0	1	1	1		final remainder $= 7/32 \cdot 2^{-5}$

- $Q=(0.100\bar{1}\bar{1})_2=13/32$ - single 0 handled properly

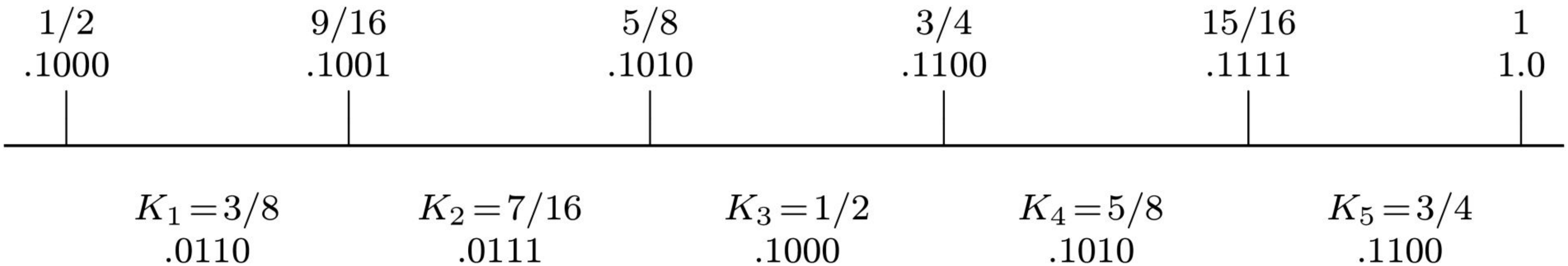
Implementing Scheme 1

- Two adders needed
  - One to add or subtract D
  - Second to add/subtract 2D if D too small (starts with 0.10 in its true form) or add/subtract D/2 if D too large (starts with 0.11)
- Output of primary adder used, unless output of alternate adder has larger normalizing shift
- Additional multiples of D possible - 3D/2 or 3D/4
- Provide higher overall average shift - about 3.7 - but more complex implementation

Modifying SRT - Scheme 2

- For $K=1/2$, ratio D/K in optimal range $3/5 \leq D \leq 3/4$ is
  $6/5 \leq D/K = D/(1/2) \leq 3/2$ or
  $(6/5)K \leq D \leq (3/2)K$
- If D not in optimal range for $K=1/2$ - choose a different comparison constant K
- Region $1/2 \leq |D| < 1$ can be divided into 5 (not equally sized) sub-regions
- Each has a different comparison constant K_i

Division into Sub-regions



- 4 bits of divisor examined for selecting comparison constant
- It has only 4 bits compared to 4 most significant bits of remainder
- Determination of sub-regions for divisor and comparison constants - numerical search
- Reason: Both are binary fractions with small number of bits to simplify division algorithm

Example

- $X=(0.00111111)_2=63/256$; $D=(0.1001)_2=9/16$
- Appropriate comparison constant - $K_2=7/16=0.0111_2$
- If remainder negative - compare to two's complement of $K_2 = 1.1001_2$

$r_0 = X$		0	.0	0	1	1	1	1	1	1	
$2r_0$		0	.0	1	1	1	1	1	1	0	≥ 0.0111 set $q_1 = 1$
Add $-D$	+	1	.0	1	1	1					
r_1		1	.1	1	1	0	1	1	1	0	
$2r_1 = r_2$		1	.1	1	0	1	1	1	0	0	≥ 1.1001 set $q_2 = 0$
$2r_2 = r_3$		1	.1	0	1	1	1	0	0	0	≥ 1.1001 set $q_3 = 0$
$2r_3$		1	.0	1	1	1	0	0	0	0	< 1.1001 set $q_4 = \bar{1}$
Add D	+	0	.1	0	0	1					
r_4		0	.0	0	0	0	0	0	0	0	zero final remainder

- $Q=0.100\bar{1}=0.0111_2=7/16$ - minimal SD form

High Radix Division

- Number of add/subtracts in radix-2 SRT is data-dependent
- Asynchronous circuit needed to use reduced number of nonzero bits in quotient
- Increasing number of 0's in quotient - limited practical significance

◆ Number of add/subtracts reduced by increasing radix β

- $\beta = 2^m$ - m quotient bits generated each step
- Number of steps reduced to $\lceil n/m \rceil$

◆ Recursive equation for remainder -

◆ $r_i = \beta \, r_{i-1} - q_i \, D$

- Multiply by $\beta=2^m$ - shift left remainder by m bit positions

◆ Digit set for quotient:

- $0,1,...,\beta-1$ for restoring division
- Up to $\overline{\beta-1},...,\overline{1},0,1,...,\beta-1$ for high-radix SRT division

High Radix Restoring Division

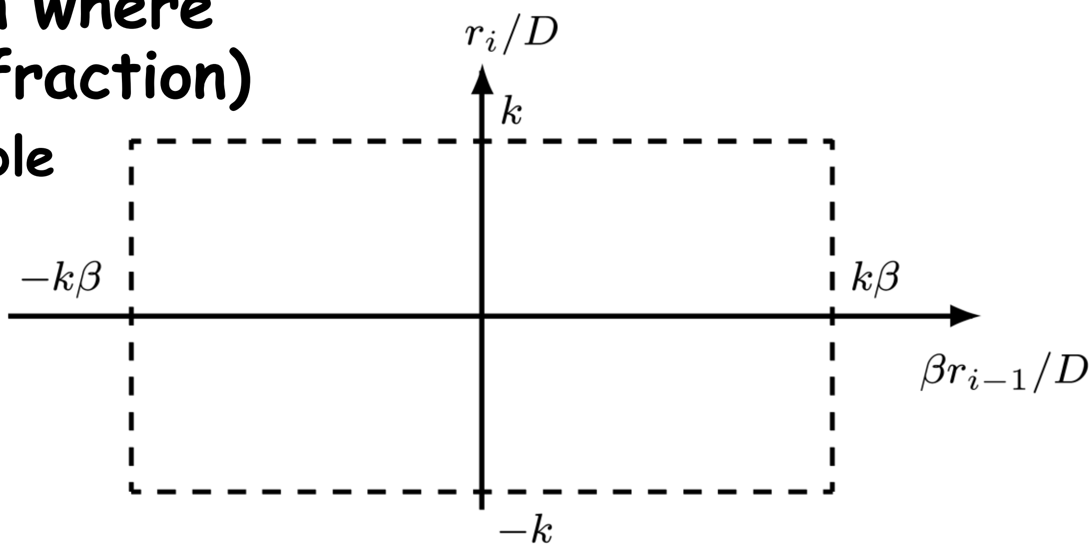
- All previous division algorithms can use radix > 2
- Restoring division -
  - Initial guess $q_i=1$
  - If remainder $\beta r_{i-1}-D>0$ - increase to $q_i=2$
  - Subtract D from temporary remainder: $\beta r_{i-1}-2D$
  - Repeat until $q_i=j$: temporary remainder $\beta r_{i-1}-jD$ negative
  - Remainder restored by adding D: $\beta r_{i-1}-(j-1)D$; $q_i=j-1$
- Time-consuming - no advantage over binary algorithm
- Can be parallelized by circuits comparing βr_{i-1} to several multiples jD - selecting smallest positive remainder; substantial hardware requirement
- Binary nonrestoring division - similar changes

High-Radix SRT Algorithm

- Faster than binary version
- Quotient digit q_i - signed digit in range $\overline{\alpha},\overline{\alpha-1},\dots,\overline{1},0,1,\dots\alpha$, where $\lceil 1/2(\beta-1)\rceil \leq \alpha \leq \beta-1$
- Finding possible choices for α in high-radix division:
- Quotient digit q_i selected so that $|r_i| < |D|$; otherwise, next quotient digit may be β or larger
  - Guarantees convergence of division procedure
  - For maximal remainder $r_{i-1}=D-ulp$ and positive D, largest value for $q_i=\alpha$ should guarantee r_i in allowable region

Reducing Remainder Range

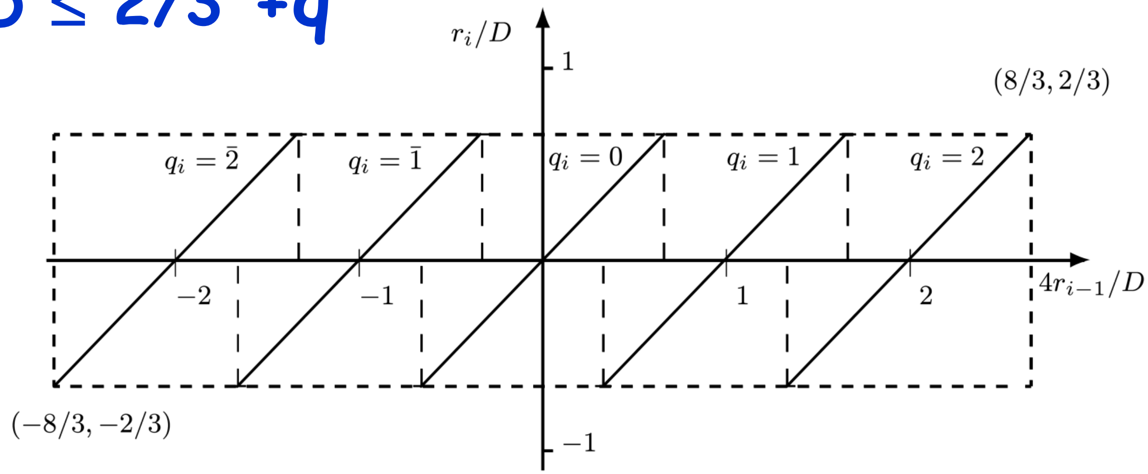
- $r_i = \beta\,(D\text{-}ulp) - \alpha D \le D\text{-}ulp$
- Select for α only maximum value $\beta - 1$
- May consider division where $|r_i| \le k|D|$, (k is a fraction)
  - reduce size of allowable region for remainder:



- $r_i = \beta\, k(D\text{-}ulp) - \alpha D \le k(D\text{-}ulp)$
- $\alpha \ge k(\beta - 1) \;\Rightarrow\; k \le \alpha/(\beta - 1)$
  - $1/2 \le k \le 1$ allows selection of any α in $\lceil(\beta-1)/2\rceil \le \alpha \le \beta - 1$
  - Larger $k \Rightarrow$ larger redundancy for quotient

Example

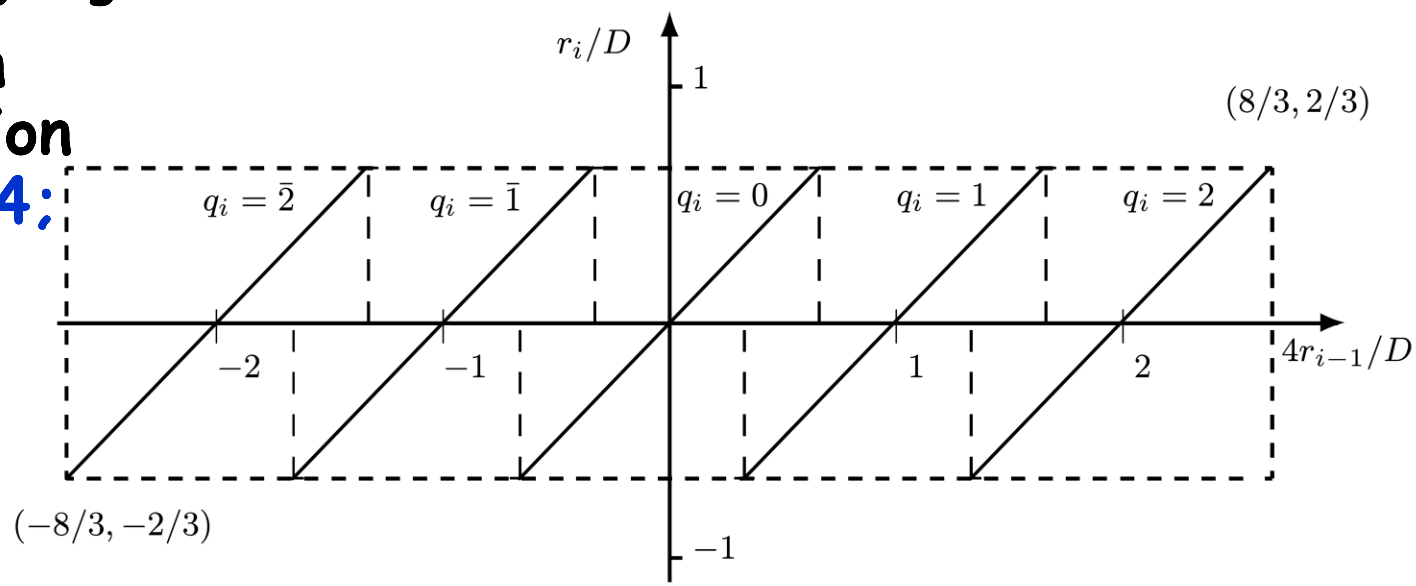
- $\beta=4$; $\alpha=2$; $k=\alpha/(\beta-1) = 2/3$
  - $|r_i| \le kD = 2/3\ D$; $|\beta r_{i-1}| = |4r_{i-1}| \le 8/3\ D$
    or $|r_i/D| \le 2/3$ and $|4r_{i-1}/D| \le 8/3$
- Digit set for $q_i = \{\bar{2},\bar{1},0,1,2\}$
- Region for selecting q : $-2/3 \le 4r_{i-1}/D - q \le 2/3$
  or $-2/3 + q \le 4r_{i-1}/D \le 2/3 + q$
- Region examples:



  - For selecting $q_i=2$:
    $4/3 \le 4r_{i-1}/D \le 8/3$
  - For selecting $q_i=1$:
    $1/3 \le 4r_{i-1}/D \le 5/3$
  - Overlapping region: $4/3 \le 4r_{i-1}/D \le 5/3$
    select either $q_i=1$ or $q_i=2$
  - Similar overlapping regions exist for any 2 adjoining digits

Measure of Redundancy

- Ratio $k=\alpha/(\beta-1)$ - measure of redundancy in representation of quotient
  - * Larger k $\Rightarrow$ larger overlap regions in plot of r_i/D vs $\beta r_{i-1}/D$
- Example: $\alpha=3$; $\beta=4$; $k=1$ - maximum redundancy
  - * Region for $q_i=1$: $0 \leq 4r_{i-1}/D \leq 2$,
  - * Region for $q_i=2$: $1 \leq 4r_{i-1}/D \leq 3$
  - * Overlapping region : $1 \leq 4r_{i-1}/D \leq 2$
- Larger than overlap region for $\alpha=2$;$\beta=4$; $k=2/3$:

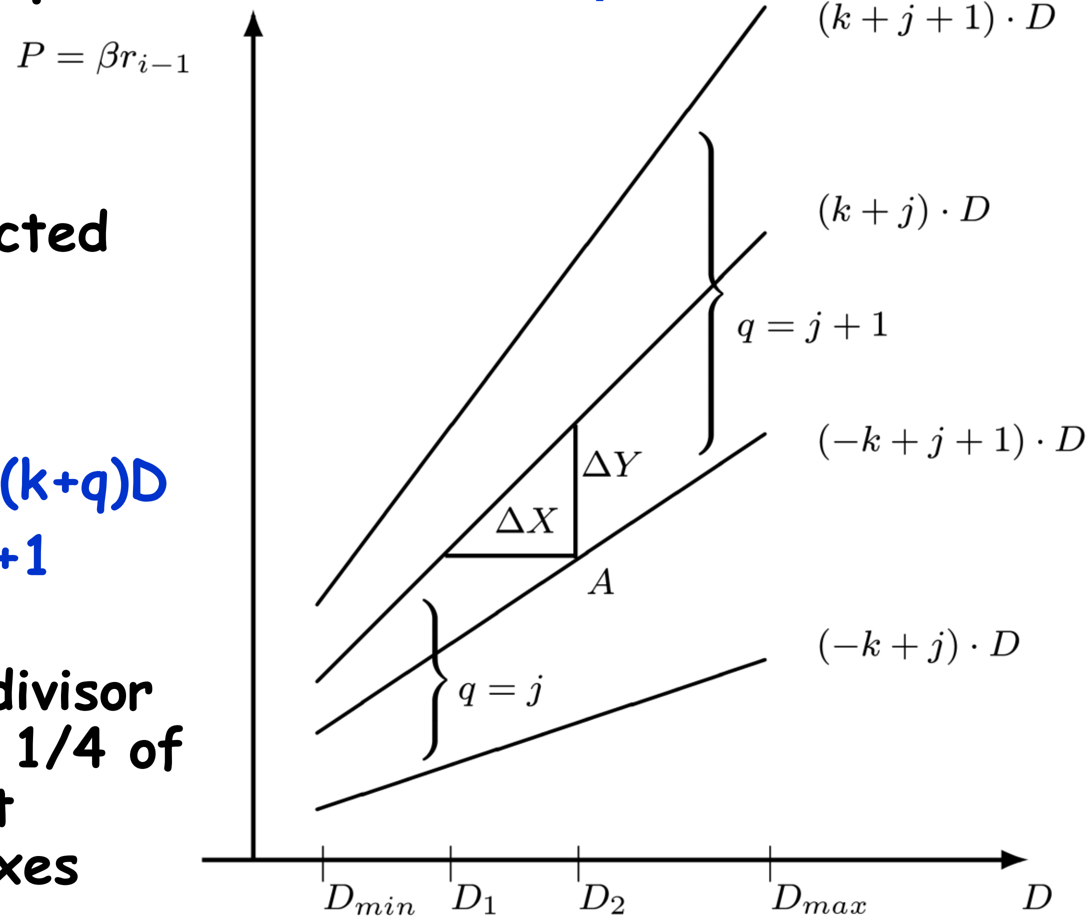


Implication of Overlap Region

- Provides choice of comparison constants for partial remainder and divisor
  - Can be selected to require as few digits as possible
  - Reducing execution time of comparison step when determining quotient digit
- Larger $\alpha \Rightarrow$ larger overlap region $\Rightarrow$ larger choice $\Rightarrow$ fewer digits
- On the other hand, larger $\alpha \Rightarrow$ more αD multiples $\Rightarrow$ extra hardware and/or time required
- For given α - determining number of bits of partial remainder and divisor to be examined is the most difficult step when developing high-radix SRT
  - Can be done numerically, analytically, graphically, or by combination of techniques

Graphical Approach: P-D Plot

- Basic equation for partial remainder - $\beta r_{i-1}=r_i+q_iD$
- Notation: P = previous partial remainder βr_{i-1}
  - Partial remainder vs. Divisor plot - indicates regions in which given values of q may be selected
  - Limits on P for given q:
  - $-kD \le r_i \le kD$
  - $P_{min}=(-k+q)D$; $P_{max}=(k+q)D$
  - Regions for $q=j$ and $q=j+1$ overlap
  - Only positive values of divisor and partial remainder - 1/4 of complete P-D plot - plot symmetric about both axes
  - Only values of $|D|$ in $[D_{min}, D_{max}]$ are of interest - e.g. [0.5,1);[1,2) (IEEE floating-point)

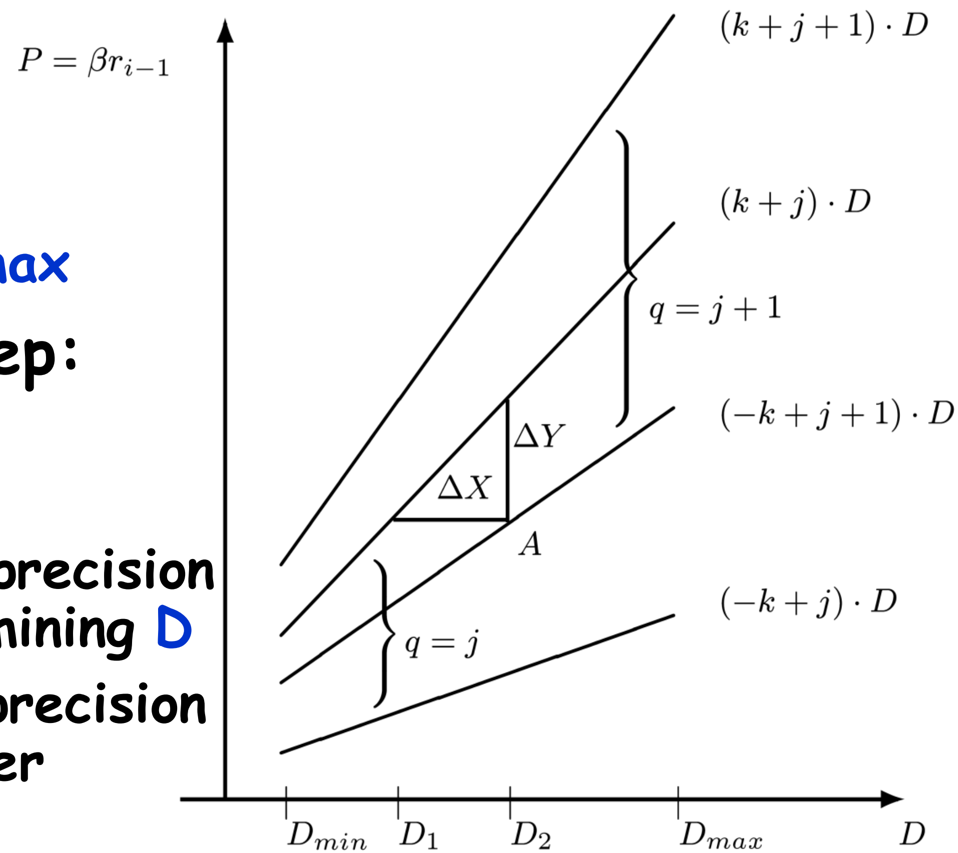


Separating Selection Regions

- Value of P separating selection regions of q=j & q=j+1
  - Serves as comparison constant
  - Its number of bits determines necessary precision when examining partial remainder to select q
- Line separating regions is horizontal (P=c ; selection of q independent of D) if and only if

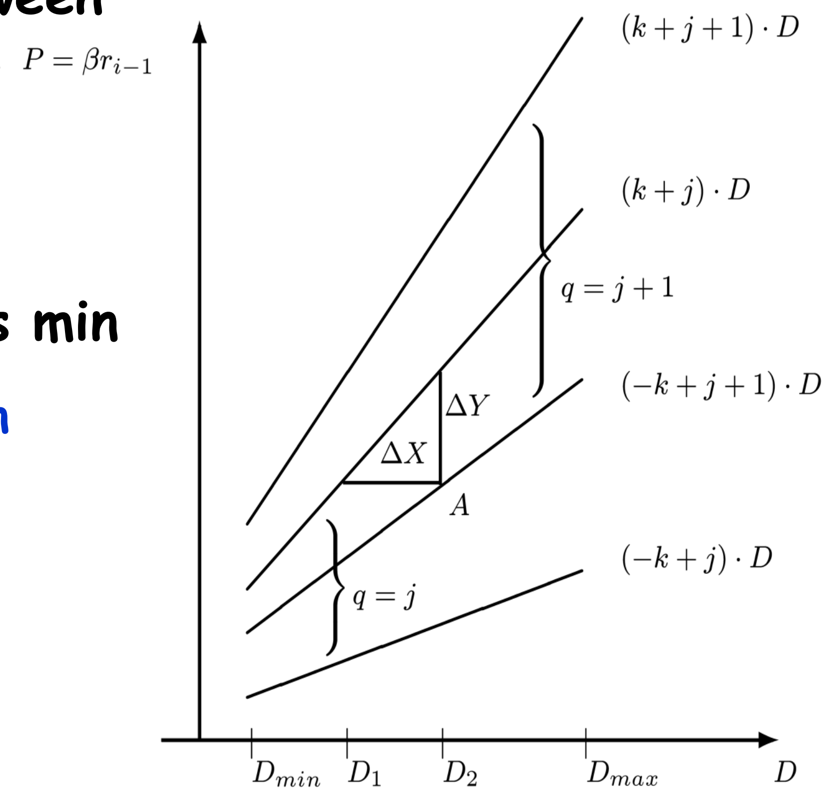
  (k+j)Dmin ≥ c ≥ (-k+j+1)Dmax

- Otherwise - line is stairstep:
  - partitioning [Dmin,Dmax) into sub-intervals
  - "Stepping" points determine precision - number of digits - of examining D
  - Height of steps determines precision of examining partial remainder



Determining Precision

- **Notation**: ΔX (ΔY) - maximum width (height) of a step between D_1 and D_2
  - Horizontal (vertical) distance between the 2 lines defining overlap region



- $\Delta X = D_2 - D_1 = P/(-k+j+1) - P/(k+j)$ $= P(2k-1)/[j(j+1)+k(1-k)]$
  - ΔX minimal when j is max and P is min
  - $j_{max} = \alpha - 1$; P minimal when $D_1 = D_{min}$
  - $\Delta X_{min} = D_{min}(k+\alpha-1)(2k-1)/[\alpha(\alpha-1)+k(1-k)]$
  - $\Delta Y = (k+j)D - (-k+j+1)D = (2k-1)D$
  - ΔY is minimal when $D = D_{min}$
- It is sufficient to consider overlapping region between $q=\alpha$ and $q=\alpha-1$ near D_{min}

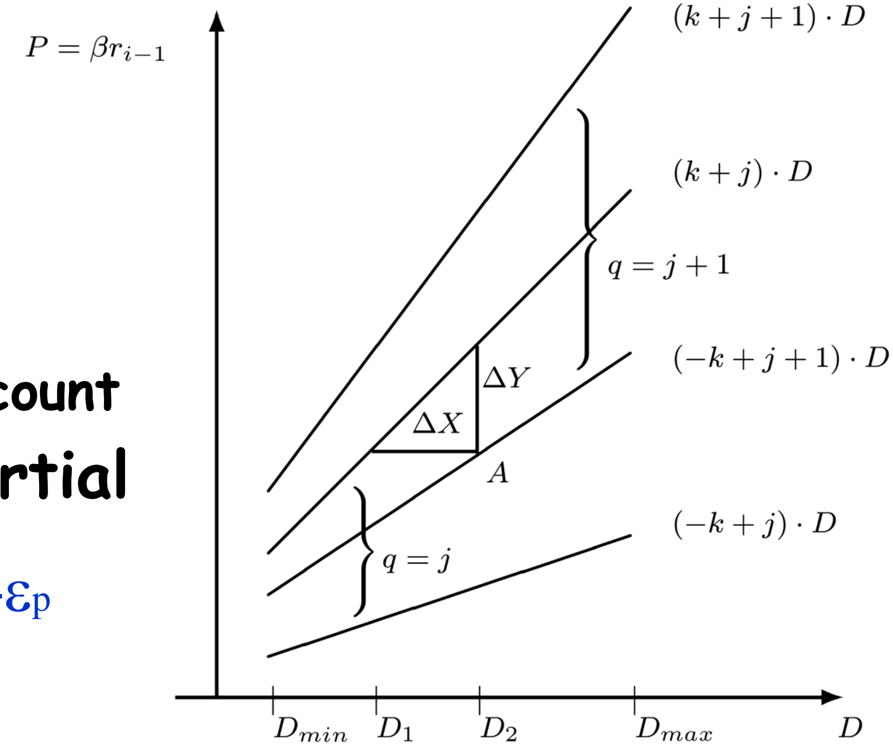
Precision - Cont.

- Notation: N_P (N_D) - number of examined bits of partial remainder (divisor) ;

  ε_P (ε_D) - number of fractional bits in N_P (N_D)
- Selecting q - look-up table implemented in a PLA (programmable logic array) with N_P+N_D inputs
- Minimizing size of look-up table speeds up division
- Precision of partial remainder ("truncated" divisor) - $2^{-\varepsilon_p}$ ($2^{-\varepsilon_D}$)
- $2^{-\varepsilon_D} \leq \Delta X_{min}$; $2^{-\varepsilon_p} \leq \Delta Y_{min}$
- Only upper bounds for precision - the 2 extreme points $\Delta X, \Delta Y$ may require higher precision - more than $\varepsilon_P, \varepsilon_D$ fractional bits

Using P-D Plot

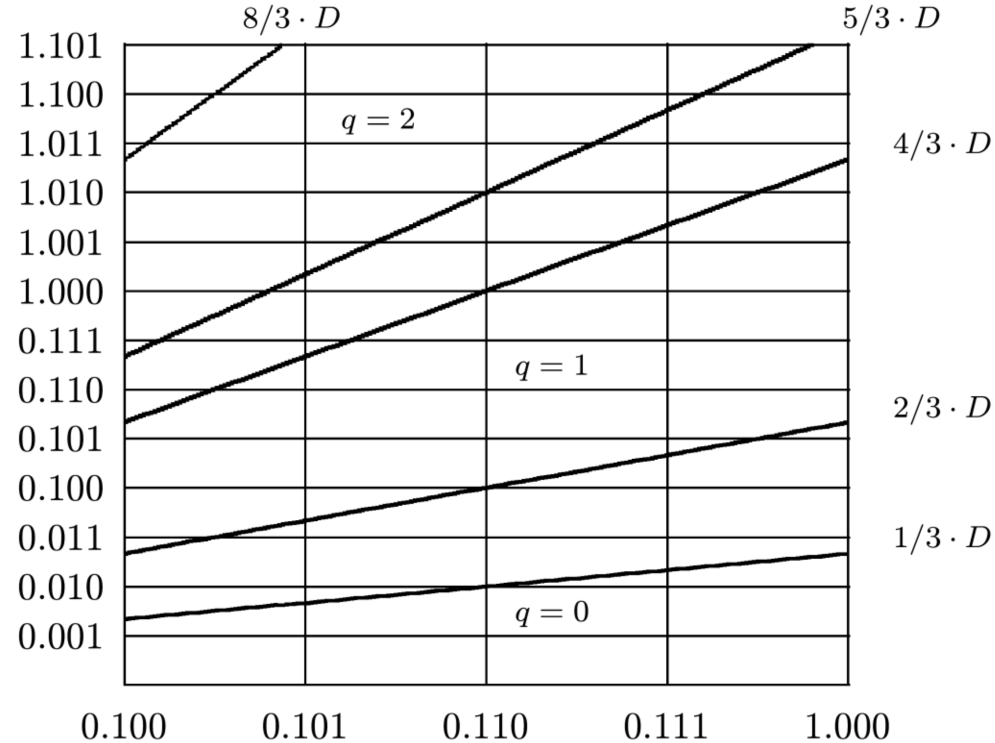
* To determine precision
* To select q for each P,D when truncated to N_P, N_D bits
* Limited precision taken into account



- Point (P,D) represents all partial remainder-divisor pairs with $P \leq$ partial remainder $\leq P+2^{-\varepsilon_P}$ $D \leq$ divisor $\leq D+2^{-\varepsilon_D}$
- Selected q must be legitimate for all pairs in range
- Example: Point A
- Divisor=D_2 - select $q=j+1$; divisor=$D_2+2^{-\varepsilon_D}$ - select $q=j$
- Conclusion: do not select $q=j+1$ for point A or any other point in overlap region whose horizontal distance from the line $(-k+j+1)D \leq 2^{-\varepsilon_D}$

Example:

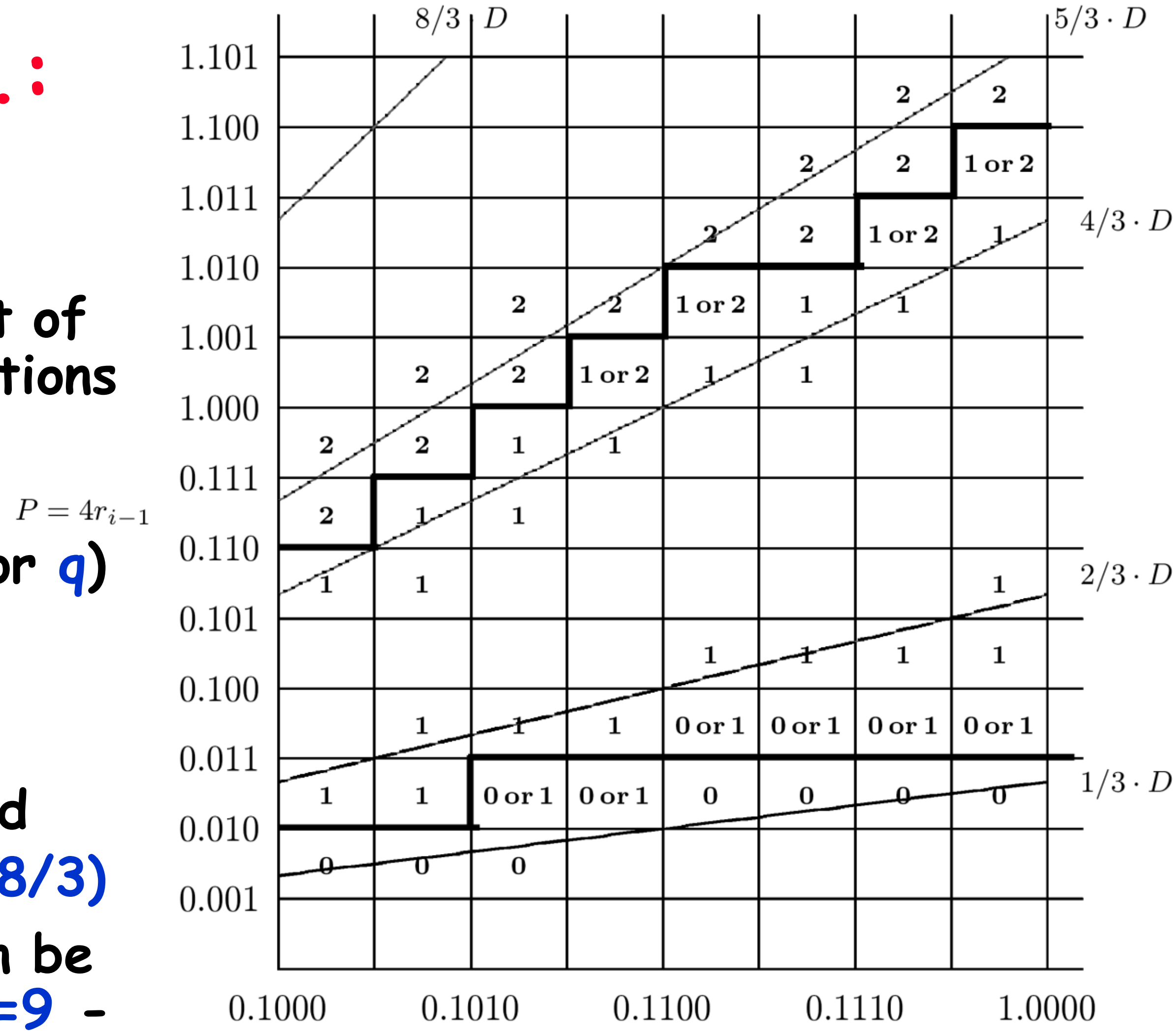
P-D Plot for β=4,α=2,D∈[0.5,1)



- Overlapping region for q=1,q=2 - between P=(k+α-1)D=5/3 D and P=(-k+α)D=4/3 D
- Single horizontal line?
  - (k+1)Dmin=5/6 < (-k+2)Dmax=4/3 - no single line
- Smallest horizontal and vertical distances:
  - ΔXmin=Dmin·5/3·3/20=1/8=2^{-3} ⇒ $\varepsilon_D \geq 3$
  - ΔYmin=Dmin·1/3=1/6 ⇒ $\varepsilon_p \geq 3$
- For pair (0.110,.100) - no q legitimate for all points in corresponding rectangle - resolved by ε_D=4

Example Cont.:

$\mathcal{E}_P=3$; $\mathcal{E}_D=4$



- Heavy lines - one out of many possible separations
- Designer can select solution to minimize PLA (look-up table for q)
- PLA has $N_D+N_P=4+6$ inputs - 3 more bits needed for integer part of remainder and its sign $(-8/3 \geq P \geq 8/3)$
- Number of inputs can be reduced to $N_P+N_D-1=9$ - most significant bit of D is always 1 - can be omitted
- $2/3 \cdot 1/2 \leq 1/3 \Rightarrow$ single line possible between q=1,q=0 - requires high-precision comparison of partial remainder - divisor interval partitioned into two subintervals

Example

- $X=(0.00111111)_2=63/256$; $D=(0.1001)_2=9/16$
- **Comparison constants -** $1/4$ (0.010) , $7/8$ (0.111)

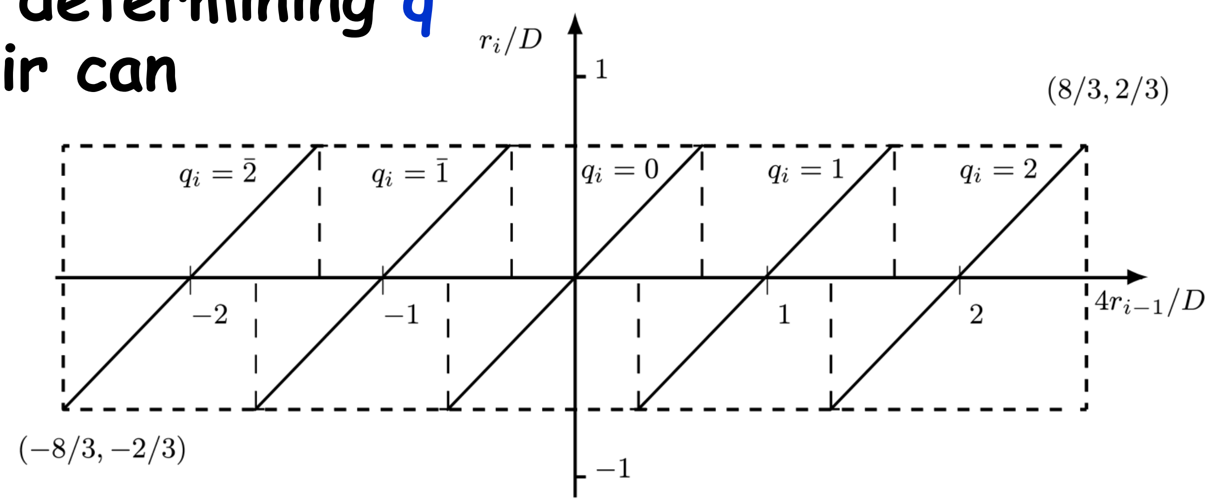
$r_0 = X$		0	.0	0	1	1	1	1	1	1	
$4r_0$	0	0	.1	1	1	1	1	1			$\geq 7/8$ set $q_1 = 2$
Add $-2D$	1	0	.1	1	1	0					
r_1	1	1	.1	1	0	1	1	1			
$4r_1$	1	1	.0	1	1	1					$< -1/4$ set $q_2 = \bar{1}$
Add D	0	0	.1	0	0	1					
r_2	0	0	.0	0	0	0					zero final remainder

- **Resulting quotient:**

$Q=0.2\bar{1}_4=0.100\bar{1}_2=0.0111_2=7/16$

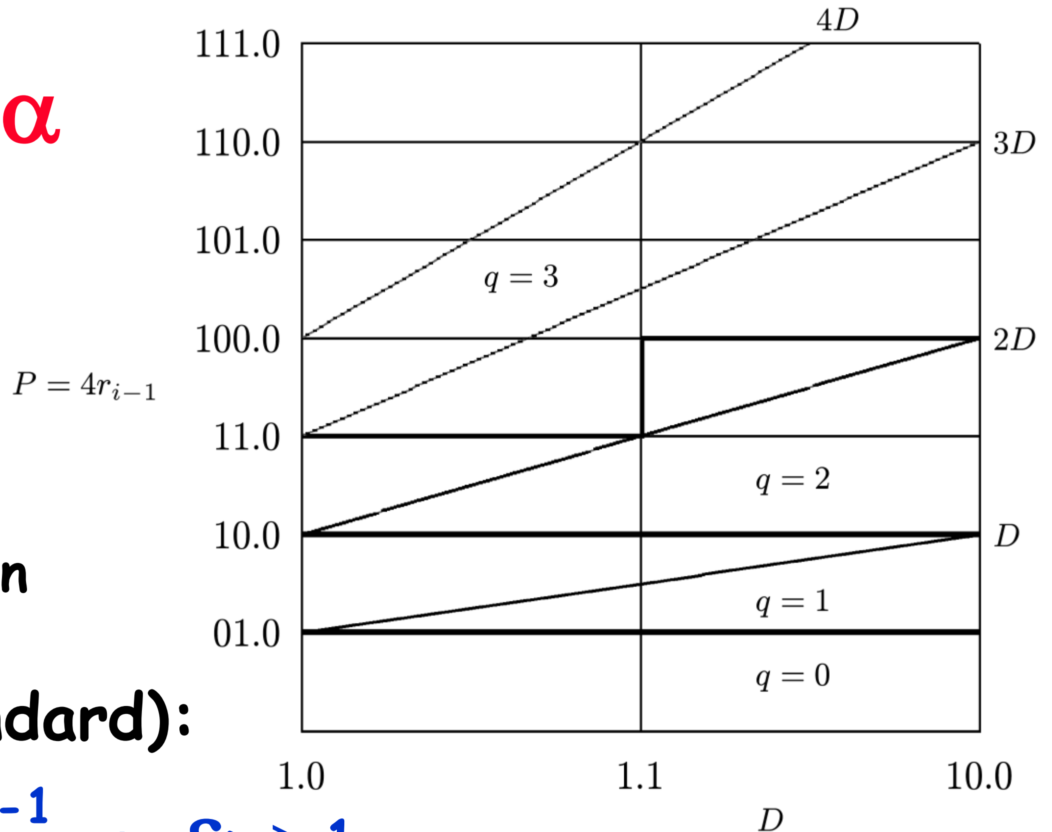
Numerical Calculation of Look Up Table

- Example- start with initial guess $\varepsilon_p=\varepsilon_D=3$ - attempt to calculate q for D=0.100 and P=0.110 (worst case)
- Divisor truncated - consider values from 0.100 to 0.101 ; partial remainder from 0.110 to 0.111
- P/D between 0.110/0.101=1.2 (q=1) and 0.111/0.100=1.75 (q=2)
- Insufficient precision - increase number of bits of either divisor or partial remainder - try again
- Numerical search determining q for each (P,D) pair can be programmed



Example - Lower Precision of Higher α

- β=4; α=3; k=α/(β-1)=1
- Region for q=2 - between P=(k+q)D=3D ; P=(-k+q)D=D
- Region for q=3 - between P=4D ; P=2D
- Overlapping region - between P=3D and P=2D
- For D$\in$[1,2) (IEEE standard):
  * ΔXmin=Dmin·3·1/6=3/6=2^{-1} $\Rightarrow$ $\varepsilon_D \geq 1$
  * ΔYmin=Dmin·1=1 $\Rightarrow$ $\varepsilon_p \geq 0$



- Based on diagram - ε_D=1 ; ε_p=0 - N_D=2 ; N_P=4 instead of N_D=4 ; N_P=6 for α=2
- Simpler quotient selection logic - costly multiple 3D

Example

- $X=(01.0101)_2=21/16$; $D=(01.1110)_2=15/8$
- Partial remainder comparison constants - 1,2,4

$r_0 = X$				0	1	.0	1	0	1	
$4r_0$		0	1	0	1	.0	1	0		≥ 4.0 set $q_1 = 3$
Add $-3D$	$+$	1	0	1	0	.0	1	1		
r_1		1	1	1	1	.1	0	1		
$4r_1$		1	1	1	0	.1	0	0		≥ -2.0 set $q_2 = \bar{1}$
Add D	$+$	0	0	0	1	.1	1	1		
r_2		0	0	0	0	.0	1	1		final remainder $= 3/8 \cdot 2^{-4}$

- Quotient: $Q=(0.3\bar{1})_4=(0.110\bar{1})_2$ = 11/16
- Verification:
  $QD+R = 11/16 \cdot 15/8 + 3/128 = 168/128 = 21/16 = X$