WASHINGTON STATE
UNIVERSITY

Dual-Purpose Hardware Algorithms and Architecture: Part II – Integer Division

Jihee Seo (Synopsys) – Presenter
Dae Hyun Kim (Washington State University)

ARITH 2023(Sep4-6, 2023)

Table of Contents

Terminology

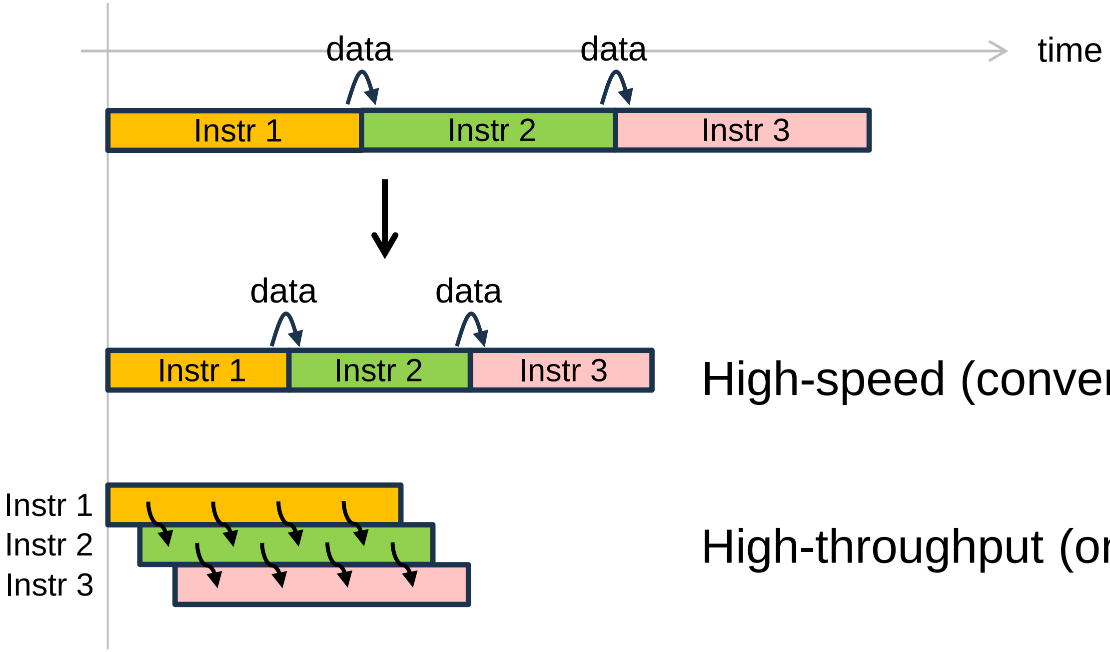
- For simplicity

- For operand :
  - Offline operand : complete operand without unknown bits.
  - Online operand : incomplete operand (e.g. 1011_1xxx)

- For divider :
  - Offline divider : process only offline operands.
  - Online divider : process both offline & online operands.

Motivation

- Target: compute-intensive applications.
  - For dependent instructions
    - Instruction 2 depends on Instruction 1
    - Instruction 3 depends on Instruction 2

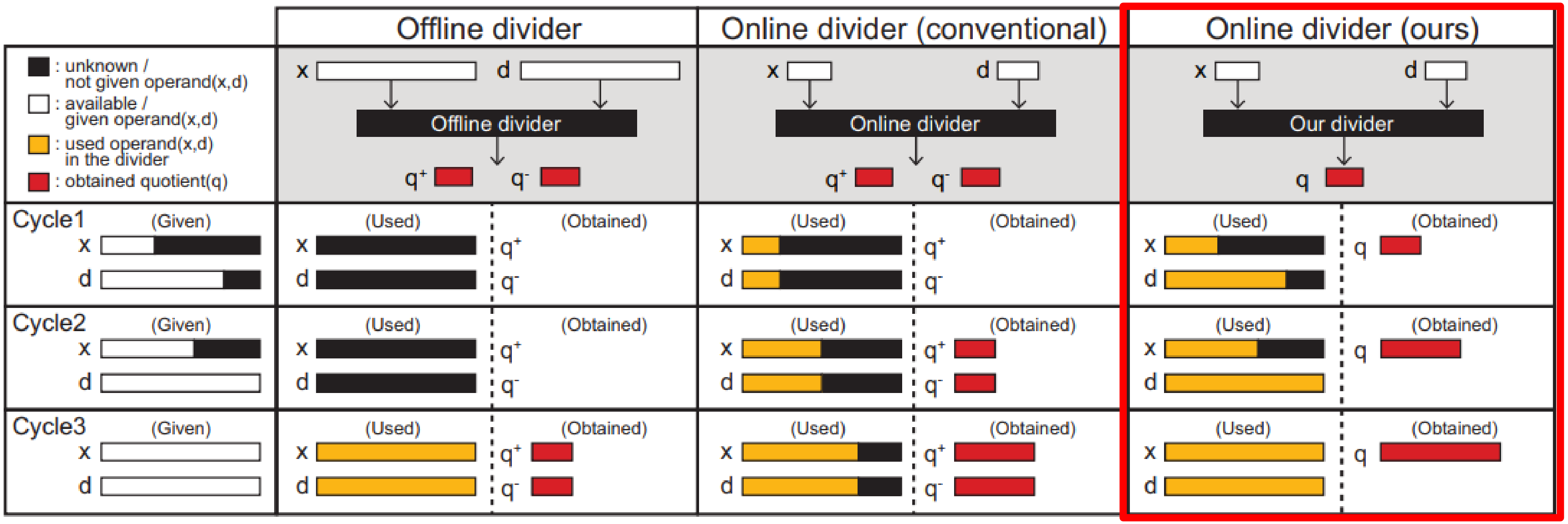


High-speed (conventional)

High-throughput (online arithmetic)

Proposed

(x: dividend / d: divisor)



(Interval analysis- based)

- Online divider(conventional) : [10] M. Ercegovac, “On-Line Arithmetic: An Overview,” in Real Time Signal Processing VII: Proc. SPIE, vol. 495, 1984, pp. 86–93

Offline Division: Digit-Recurrence (Review)

- Notation (integer)
  - Quotient obtained down to the $(n-j)$-th digit (j-th iteration)
    - $q[j] = q_{n-1}\underbrace{q_{n-2} \ldots q_{n-j}(\mathrm{xx} \ldots}_{\text{obtained}}\underbrace{\mathrm{x})}_{\text{to be obtained}} = \sum_{i=1}^{j} q_{n-i} \cdot r^{n-i}$

- Digit-recurrence division
  - Division: $x = d \cdot q + rem$
  - Condition: $|rem| < |d| \cdot ulp$
  - At j-th iteration

$$rem[j] = x - d \cdot q[j]$$

$$\begin{aligned} & 0 \leq rem < d \cdot r^{n-j} \\ \Leftrightarrow\ & 0 \leq x - d \cdot q[j] < d \cdot r^{n-j} \\ \Leftrightarrow\ & 0 \leq \underbrace{r^{j-n} \cdot (x - d \cdot q[j])}_{w[j]} < d \end{aligned}$$

Offline Division: Digit-Recurrence (Review)

- Digit-recurrence division
  - At j-th iteration

$$0 \le \underbrace{r^{j-n} \cdot (x - d \cdot q[j])}_{w[j] \text{ (partial remainder)}} < d$$

  - At $(j+1)$-th iteration

$$q[j+1] = \underbrace{q_{n-1}q_{n-2} \dots q_{n-j}}_{\text{obtained}} \underset{\substack{\uparrow \\ \text{Find}}}{q_{n-j-1}}(\text{xx} \dots \text{x})$$

$$0 \le w[j+1] < d$$

$$\Leftrightarrow 0 \le r \cdot w[j] - d \cdot q_{n-j-1} < d$$

  - $q_{n-j-1} = k$ if

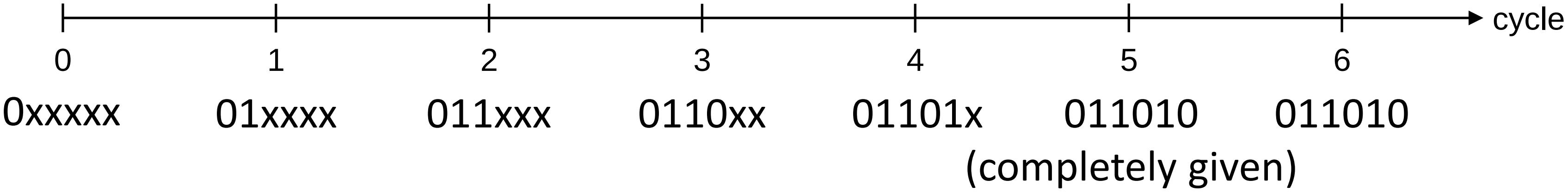
$$0 \le r \cdot w[j] - d \cdot k < d$$

$$\Leftrightarrow \boxed{k \cdot d \le r \cdot w[j] < (k+1) \cdot d}$$

Radix-r digit-recurrence division

Online Arithmetic: Review

- Operands
  - Given one digit per cycle.
  - Example: $x = 011010_2$



0	1	2	3	4	5	6	cycle
0xxxxx	01xxxx	011xxx	0110xx	01101x	011010 (completely given)	011010	

- Principles
  - Find the result with incomplete operands.

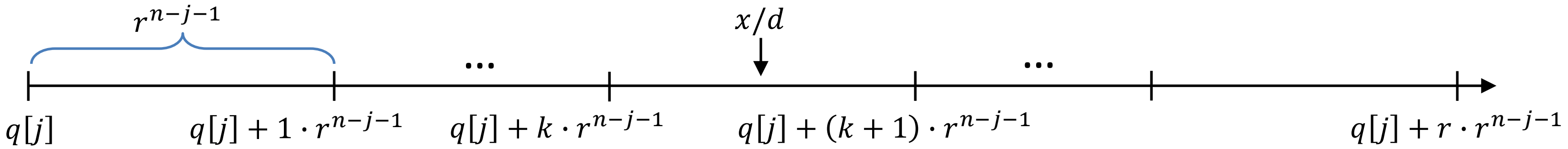
Offline Division: Interval Analysis

- Interval-analysis-based division
  - Quotient obtained down to the $(n-j)$-th digit (j-th iteration)
    - $q[j] = q_{n-1}\underbrace{q_{n-2} \dots q_{n-j}}_{\text{obtained}}\underbrace{(\mathrm{xx}\dots\mathrm{x})}_{\text{to be obtained}} = \sum_{i=1}^{j} q_{n-i} \cdot r^{n-i}$

$$q[j+1] = \underbrace{q_{n-1}q_{n-2} \dots q_{n-j}}_{\text{obtained}}\underset{\substack{\uparrow \\ \text{Find}}}{q_{n-j-1}}(\mathrm{xx}\dots\mathrm{x})$$

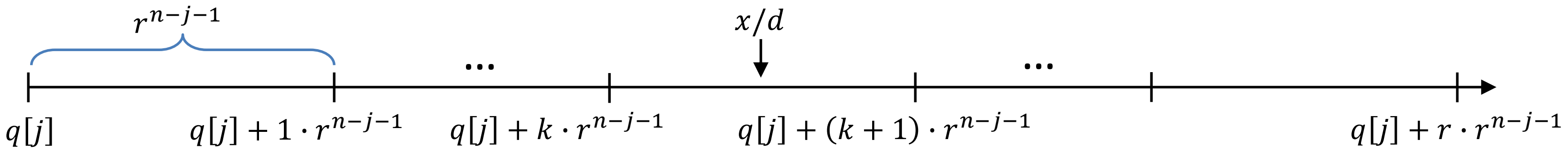
  - Condition for $q_{n-j-1} = k$

$$q[j] + k \cdot r^{n-j-1} \le \frac{x}{d} < q[j] + (k+1) \cdot r^{n-j-1}$$



Offline Division: Interval Analysis

- Interval-analysis-based division
  - Condition



$$q_{n-j-1} = k$$

$$\Leftrightarrow \; q[j] + k \cdot r^{n-j-1} \leq \frac{x}{d} < q[j] + (k+1) \cdot r^{n-j-1}$$

$$\Leftrightarrow \; d \cdot \left(q[j] + k \cdot r^{n-j-1}\right) \leq x < d \cdot \left(q[j] + (k+1) \cdot r^{n-j-1}\right)$$

$$\Leftrightarrow \; d \cdot k \cdot r^{n-j-1} \leq x - d \cdot q[j] < d \cdot (k+1) \cdot r^{n-j-1}$$

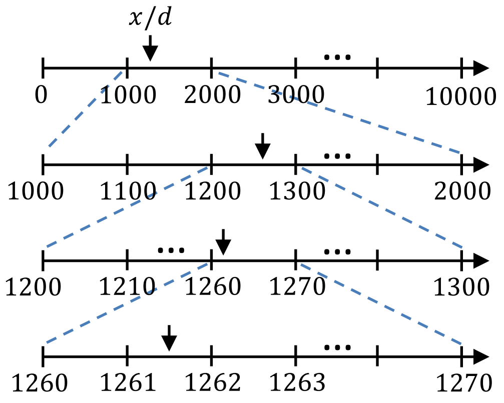
$$\Leftrightarrow \; \boxed{k \cdot d \leq r^{j+1-n} \cdot (x - d \cdot q[j]) < (k+1) \cdot d}$$

Radix-r digit-recurrence division

Offline Division: Interval Analysis

- Example (decimal numbers)
  - x (dividend): 3784
  - d (divisor): 3
  - q (quotient): xxxx

Division	Implementation
– $1000 \leq \frac{x}{d} \leq 1999 \rightarrow q$ = 1xxx	$1000 \cdot d \leq x < 2000 \cdot d$
– $1200 \leq \frac{x}{d} \leq 1299 \rightarrow q$ = 12xx	$1200 \cdot d \leq x < 1300 \cdot d$
– $1260 \leq \frac{x}{d} \leq 1269 \rightarrow q$ = 126x	$1260 \cdot d \leq x < 1270 \cdot d$
– $1261 \leq \frac{x}{d} \leq 1261 \rightarrow q$ = 1261	$1261 \cdot d \leq x < 1262 \cdot d$

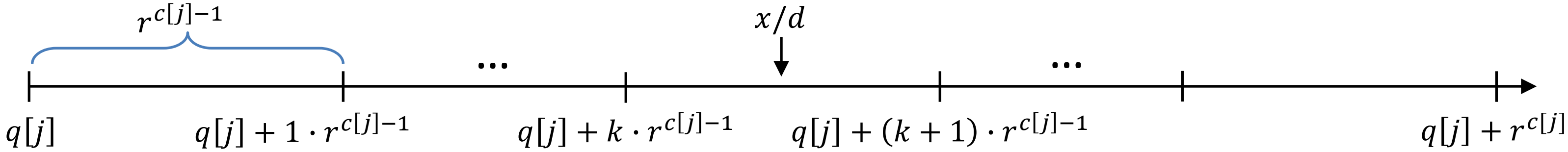


Online Division

- Input (given until the j-th iteration)
  - Dividend: $x[j] = \underbrace{x_{n-1}x_{n-2} \dots x_{a[j]}}_{\text{given}}\underbrace{(\mathrm{xx} \dots \mathrm{x})}_{\text{to be given}} = \sum_{i=a[j]}^{n-1} x_i \cdot r^i$
  - Divisor: $d[j] = \underbrace{d_{n-1}d_{n-2} \dots d_{b[j]}}_{\text{given}}\underbrace{(\mathrm{xx} \dots \mathrm{x})}_{\text{to be given}} = \sum_{i=b[j]}^{n-1} d_i \cdot r^i$
- Output (obtained until the j-th iteration)
  - Quotient: $q[j] = \underbrace{q_{n-1}q_{n-2} \dots q_{c[j]}}_{\text{obtained}}\underbrace{(\mathrm{xx} \dots \mathrm{x})}_{\text{to be obtained}} = \sum_{i=c[j]}^{n-1} q_i \cdot r^i$

Online Division: Interval Analysis

- Given
  - $x[j] = x_{n-1}x_{n-2} \dots x_{a[j]}(\mathrm{xx} \dots \mathrm{x})$
  - $d[j] = d_{n-1}d_{n-2} \dots d_{b[j]}(\mathrm{xx} \dots \mathrm{x})$
- Obtained
  - $q[j] = q_{n-1}q_{n-2} \dots q_{c[j]}(\mathrm{xx} \dots \mathrm{x})$
- Find $q_{c[j]-1}$



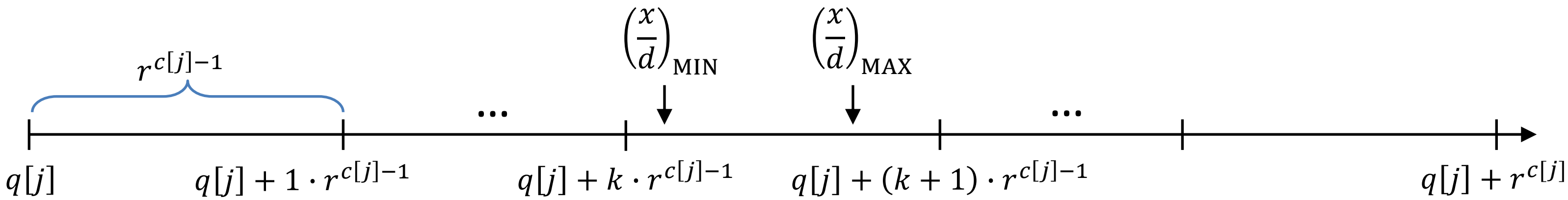
- Since x and d are partially given, we cannot find $\frac{x}{d}$.
  - But we can find the range of $\frac{x}{d}$.

Online Division: Interval Analysis

- Given
  - $x[j] = x_{n-1}x_{n-2} \dots x_{a[j]}(\mathrm{xx} \dots \mathrm{x})$
    - $x_{\mathrm{MIN}} = x_{n-1}x_{n-2} \dots x_{a[j]}0 \dots 0$ (padd 0’s)
    - $x_{\mathrm{MAX}} = x_{n-1}x_{n-2} \dots x_{a[j]}1 \dots 1$ (padd 1’s)
  - $d[j] = d_{n-1}d_{n-2} \dots d_{b[j]}(\mathrm{xx} \dots \mathrm{x})$
    - $d_{\mathrm{MIN}} = d_{n-1}d_{n-2} \dots d_{b[j]}0 \dots 0$
    - $d_{\mathrm{MAX}} = d_{n-1}d_{n-2} \dots d_{b[j]}1 \dots 1$
- Range of $\frac{x}{d} = \left[\left(\frac{x}{d}\right)_{\mathrm{MIN}}, \left(\frac{x}{d}\right)_{\mathrm{MAX}}\right]$
  - $\left(\frac{x}{d}\right)_{\mathrm{MIN}} = \frac{x_{\mathrm{MIN}}}{d_{\mathrm{MAX}}} = \frac{x_{n-1}x_{n-2} \dots x_{a[j]}0 \dots 0}{d_{n-1}d_{n-2} \dots d_{b[j]}1 \dots 1}$
  - $\left(\frac{x}{d}\right)_{\mathrm{MAX}} = \frac{x_{\mathrm{MAX}}}{d_{\mathrm{MIN}}} = \frac{x_{n-1}x_{n-2} \dots x_{a[j]}1 \dots 1}{d_{n-1}d_{n-2} \dots d_{b[j]}0 \dots 0}$

Online Division: Interval Analysis

- Obtained
  - $q[j] = q_{n-1}q_{n-2} \dots q_{c[j]}(\mathrm{xx} \dots \mathrm{x})$
- Find $q_{c[j]-1}$
  - $q_{c[j]-1} = k$ if

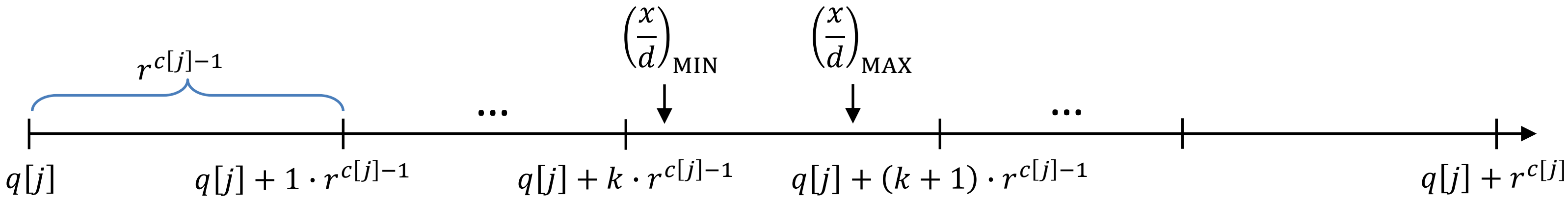


  - This just means
    - $q_{n-1}q_{n-2} \dots q_{c[j]}k0 \dots 0 \le \frac{x}{d} < q_{n-1}q_{n-2} \dots q_{c[j]}(k+1)0 \dots 0$
    - which means $q[j] = q_{n-1}q_{n-2} \dots q_{c[j]}\boldsymbol{k} \dots$
  - For example, for $r = 4$
    - $\underbrace{0321}_{\text{obtained}}\mathbf{2} \dots \le \frac{x}{d} < \underbrace{0321}_{\text{obtained}}\mathbf{3} \dots \Rightarrow \frac{x}{d} = 0321\mathbf{2} \dots$

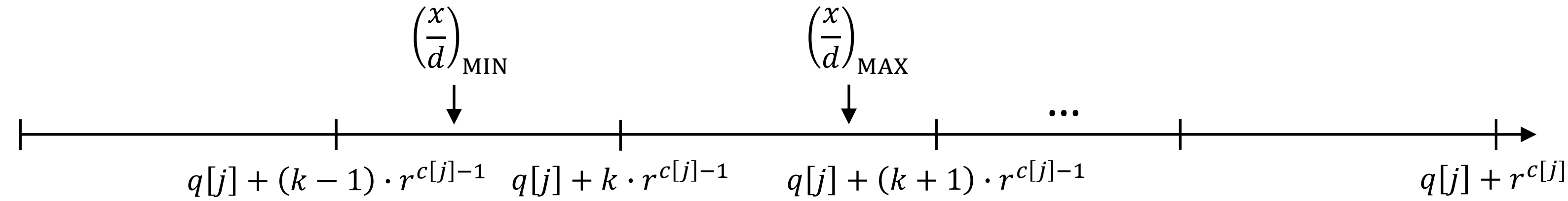
$q_{c[j]-1} = 2$

Online Division: Interval Analysis

- Case 1: $q_{c[j]-1} = k$



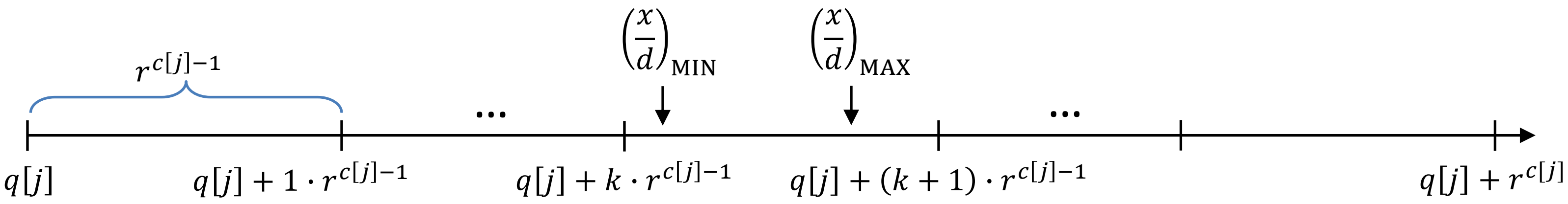
- Case 2: $q_{c[j]-1} =?$



  - In the example above, $q_{c[j]-1}$ could be $k-1$ or k.
  - We need more digits of x and/or d to find $q_{c[j]-1}$.

Online Division: Interval Analysis

- Formula: Two inequalities



Ineq#1) $q[j]+\mathrm{k}\cdot \mathrm{r}^{c[j]-1} \leq \left(\frac{\mathrm{x}}{\mathrm{d}}\right)_{\mathrm{MIN}}$

$$\rightarrow\ q[j]+\mathrm{k}\cdot \mathrm{r}^{c[j]-1} \leq \frac{x_{n-1}x_{n-2}\dots x_{a[j]}0\dots 0}{d_{n-1}d_{n-2}\dots d_{b[j]}1\dots 1} = \frac{x[j]}{d[j]+r^{b[j]}-1}$$

Ineq#2) $\left(\frac{\mathrm{x}}{\mathrm{d}}\right)_{\mathrm{MAX}} < q[j]+(\mathrm{k}+1)\cdot \mathrm{r}^{c[j]-1}$

$$\rightarrow\ \frac{x_{n-1}x_{n-2}\dots x_{a[j]}1\dots 1}{d_{n-1}d_{n-2}\dots d_{b[j]}0\dots 0} = \frac{x[j]+r^{a[j]}-1}{d[j]} < q[j]+(k+1)\cdot r^{c[j]-1}$$

Online Division: Interval Analysis

1. $q[j] + k \cdot r^{c[j]-1} \leq \frac{x[j]}{d[j]+r^{b[j]}-1}$ (Solve it for $k = 0,1,\dots,r-1$)

$\Leftrightarrow \left(q[j] + k \cdot r^{c[j]-1}\right) \cdot \left(d[j] + r^{b[j]} - 1\right) \leq x[j]$ Inequality 1

2. $\frac{x[j]+r^{a[j]}-1}{d[j]} < q[j] + (k+1) \cdot r^{c[j]-1}$ (Solve it for $k = 0,1,\dots,r-1$)

$\Leftrightarrow x[j] + r^{a[j]} - 1 < d[j] \cdot \left(q[j] + (k+1) \cdot r^{c[j]-1}\right)$ Inequality 2

- What if x and d are offline (fully given at time 0)?
  - Then, $a[j] = b[j] = 0$ for all j and $c[j] = n - j$.

1. $\left(q[j] + k \cdot r^{n-j-1}\right) \cdot d[j] \leq x[j]$
2. $x[j] < d[j] \cdot \left(q[j] + (k+1) \cdot r^{n-j-1}\right)$

$$\Rightarrow k \cdot d \leq r^{j+1-n} \cdot (x - d \cdot q[j]) < (k+1) \cdot d$$

Radix-r digit-recurrence division

In this case, the online division algorithm is equivalent to the offline division algorithm. (dual-purpose)

Online Division: Interval Analysis

1. Inequality 1 (min)

$$\left(q[j] + k \cdot r^{c[j]-1}\right) \cdot \left(d[j] + r^{b[j]} - 1\right) \leq x[j]$$

$$\Leftrightarrow \underbrace{q[j] \cdot d[j]}_{\boldsymbol{m}} + \underbrace{q[j] \cdot r^{b[j]}}_{\text{shift } q[j]} - q[j] + \underbrace{k \cdot d[j] \cdot r^{c[j]-1}}_{\text{shift } k \cdot d[j]} + \underbrace{k \cdot d[j] \cdot r^{b[j]+c[j]-1}}_{\text{shift } k \cdot d[j]} - \underbrace{k \cdot r^{c[j]-1}}_{\text{shift } k} \leq x[j]$$

2. Inequality 2 (max)

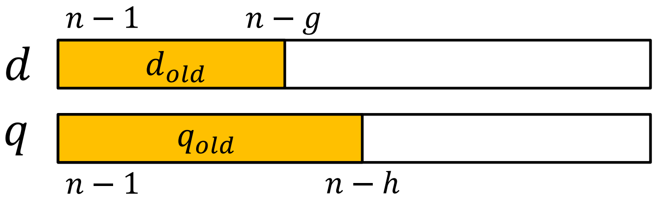
$$x[j] + r^{a[j]} - 1 < d[j] \cdot \left(q[j] + (k+1) \cdot r^{c[j]-1}\right)$$

$$\Leftrightarrow x[j] + r^{a[j]} - 1 < \underbrace{q[j] \cdot d[j]}_{\boldsymbol{m}} + \underbrace{(k+1) \cdot d[j] \cdot r^{c[j]-1}}_{\text{shift } (k+1) \cdot d[j]}$$

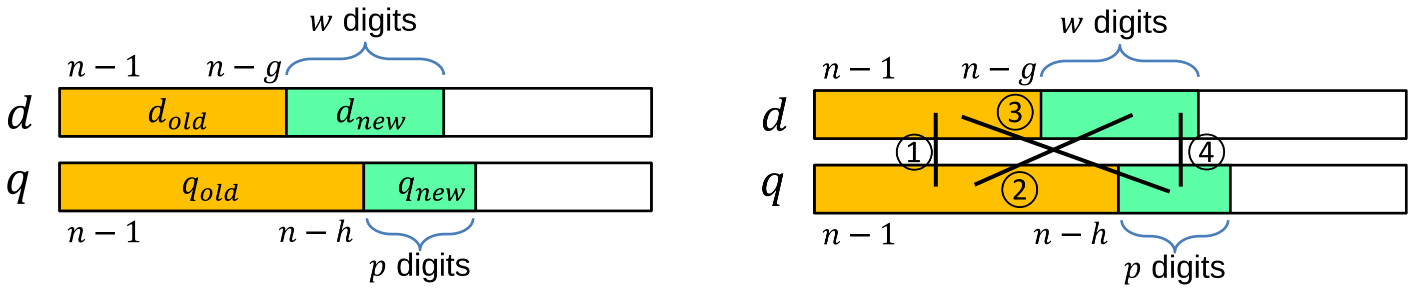
- $m[g,h] = \left(d_{n-1} \dots d_{n-g} 0 \dots 0\right) \cdot \left(q_{n-1} \dots q_{n-h} 0 \dots 0\right)$
  - We can incrementally update $m[g,h]$.

Online Division: Interval Analysis

- Incremental update of $m[g,h] = (d_{n-1} \ldots d_{n-g} 0 \ldots 0) \cdot (q_{n-1} \ldots q_{n-h} 0 \ldots 0)$.
  - Suppose we have obtained $m[g,h]$.



  - A few more digits are given (d: w more digits. q: p more digits).



  - $m[g+w,h+p] = m[g,h]$ ⋯ ① $d_{old} * q_{old}$

$+ (d_{n-g-1:n-g-w} \cdot r^{n-g-w} \cdot q_{n-1:n-h} \cdot r^{n-h})$ ⋯ ② $d_{new} * q_{old}$

$+(d_{n-1:n-g} \cdot r^{n-g} \cdot q_{n-h-1:n-h-p} \cdot r^{n-h-p})$ ⋯ ③ $d_{old} * q_{new}$

$+(d_{n-g-1:n-g-w} \cdot r^{n-g-w} \cdot q_{n-h-1:n-h-p} \cdot r^{n-h-p})$ ⋯ ④ $d_{new} * q_{new}$

If w and p are small,
this update can be
done in a cycle.
➔ Use w, p: 2 and 4bit

Integer Division

- Speed-up technique

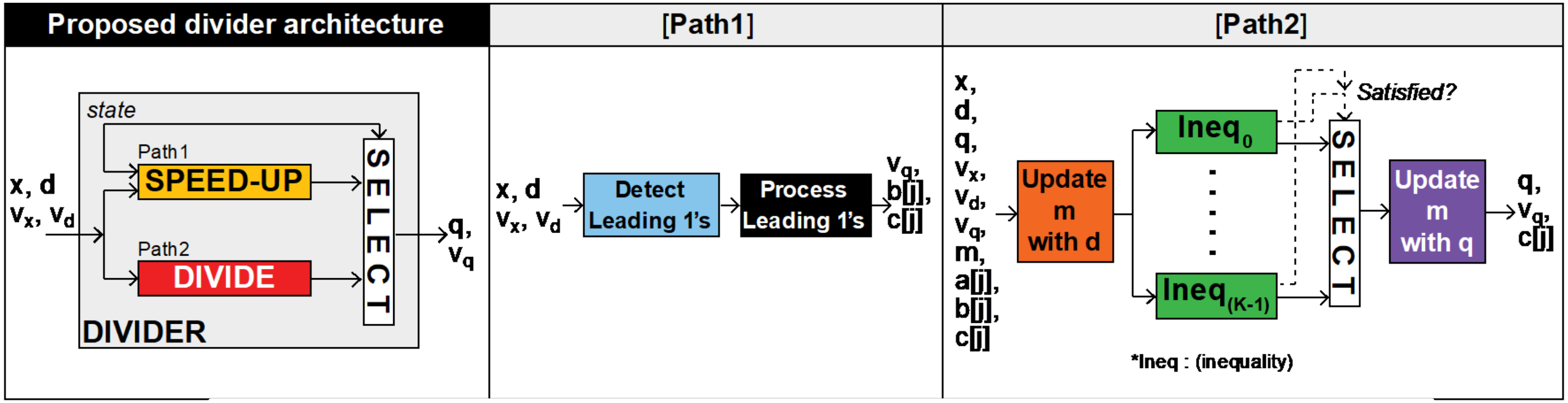
- Leading 1 locations
  - x: l_x
  - d: l_d

- If $l_x < l_d$: $q = 0$

- If $l_x \geq l_d$: $q = \underbrace{00\ldots0}_{l_x - l_d + 1 \text{ digits}}(\mathrm{xx}\ldots\mathrm{x})$

Hardware Architecture

- Top-level view

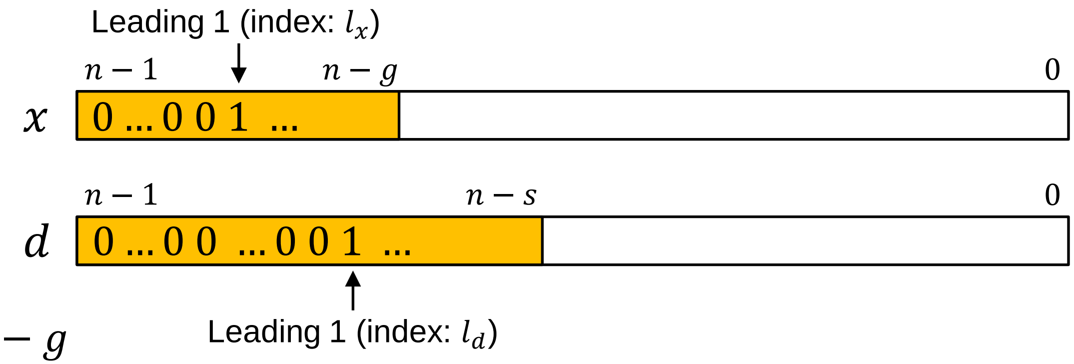


Path 1: Leading 1 process

Path 2: Actual division

Hardware Architecture

- Path 1: Detect and process leading 1's in x and d
  - If x or d does not have a leading 1, stay in Path 1.
  - If both x and d have leading 1's, then



- $a[j] = n - g$
- $b[j] = \left\lceil \frac{n-s}{w} \right\rceil$
- $c[j] = \left(\left\lceil \frac{l_x}{\log_2 r} \right\rceil - \left\lceil \frac{l_d}{\log_2 r} \right\rceil + 1 \right) \cdot \log_2 r$
- $m\big[b[j], c[j]\big] = 0$
- then move on to Path 2.

Hardware Architecture

- Path 2: Find the remaining quotient bits by the inequalities.

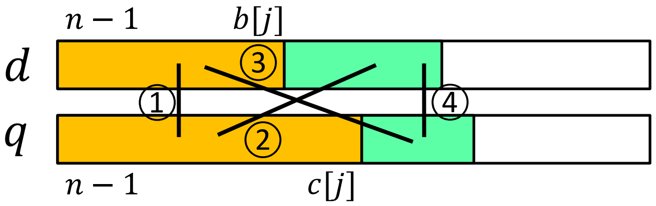
1. Update $m\big[b[j], c[j]\big]$ if $d_{b[j]-1:b[j]-w}$ is available.

2. Compute the following inequalities for $k = 0,1, \dots, r-1$.
    1. $\left(q[j] + k \cdot r^{c[j]-1}\right) \cdot \left(d[j] + r^{b[j]} - 1\right) \le x[j]$
    2. $x[j] + r^{a[j]} - 1 < d[j] \cdot \left(q[j] + (k+1) \cdot r^{c[j]-1}\right)$
    - Use carry-save adders (CSAs) to reduce the delay.

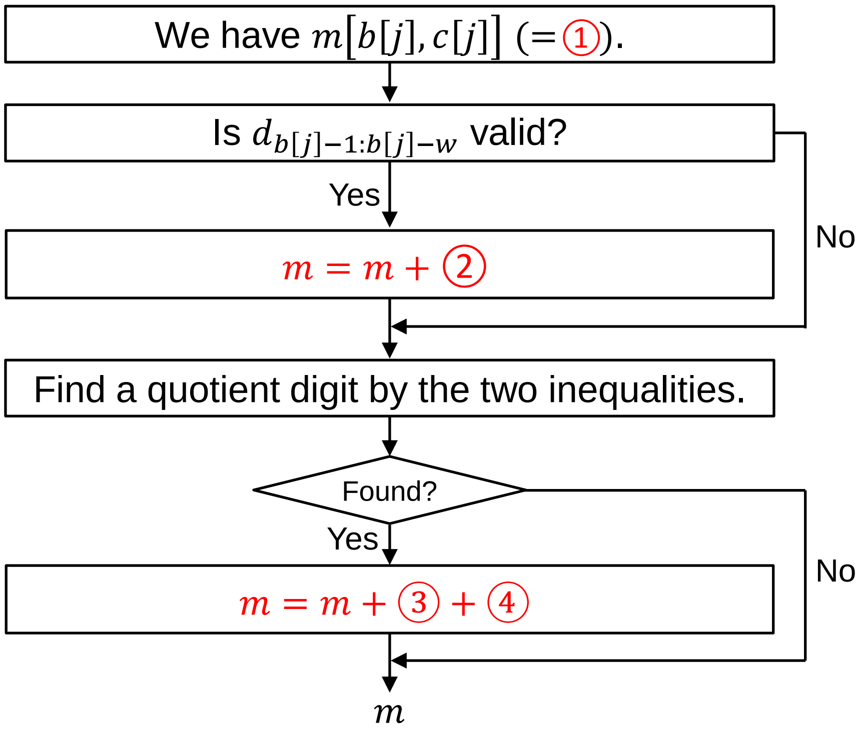
3. If the inequalities are true for a certain k, then
    1. $q_{c[j]-1} = k$
    2. Update m.

Incremental update

- When do we perform the incremental update of $m = d \cdot q$?
  1. When new digits of d are given.
  2. When we find new digits of q.



- Flowchart


We have $m[b[j],c[j]]$ (=①).

Is $d_{b[j]-1:b[j]-w}$ valid?

Yes

No

$m = m + ②$

Find a quotient digit by the two inequalities.

Found?

Yes

No

$m = m + ③ + ④$

m

Four cases

New valid digits of d	Found new Digit of q	Updated m
No	No	$m[b[j],c[j]]$
No	Yes	$m[b[j],c[j]-1]$
Yes	No	$m[b[j]-w,c[j]]$
Yes	Yes	$m[b[j]-w,c[j]-1]$

Simulation Setup

- **Design**
  - 64bit divider (64bit dividend & divisor)

- **Implementation**
  - Verilog

- **Synthesis**
  - Synopsys Design Compiler
  - 22nm standard cell library

- **Execution time simulation**
  - C/C++

- **Performance metrics**
  - Clock period, Area, Energy
  - Execution time

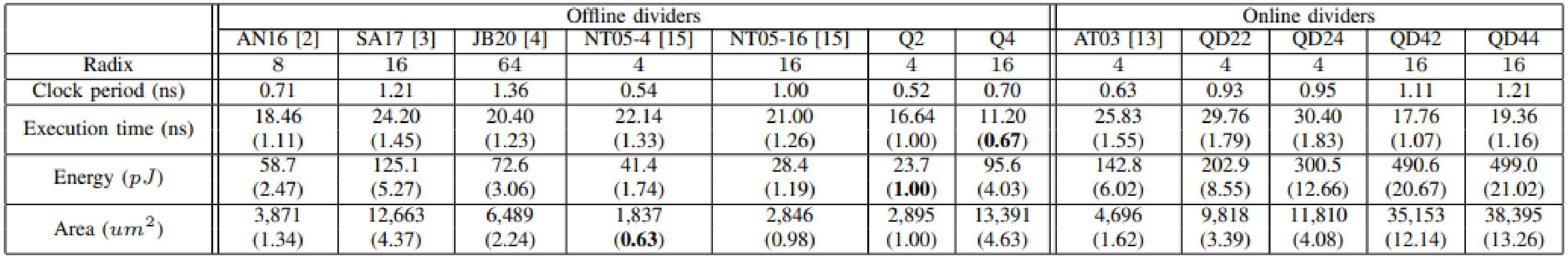
Design characteristics

	Design	Type	Radix	#divisor bits given / cycle	Description
Others	AN16	Offline	8	-	FP
	SA17	Offline	16	-	FP
	JB20	Offline	64	-	ARM, FP
	NT05	Offline	4, 16	-	Signed-digit based integer divider
	AT03	Online	4	2	The latest on-line divider, FP
Ours	Q2	Offline	4	-	Interval-analysis-based offline divider (not based on partial-remainder $w[j]$)
	Q4	Offline	16	-	
	QD22	Online	4	2	
	QD24	Online	4	4	
	QD42	Online	16	2	
	QD44	Online	16	4	

- AN16 : A. Nannarelli, “Performance/Power Space Exploration for Binary64 Division Units,” in IEEE Trans. on Computers, vol. 65, no. 5, May 2016, pp. 1671–1677
- SA17 : S. Amanollahi and G. Jaberipur, “Energy-Efficient VLSI Realization of Binary64 Division with Redundant Number Systems,” in IEEE Trans. on VLSI Systems, vol. 25, no. 3, Mar. 2017, pp. 954–961
- JB20 : J. D. Bruguera, “Low Latency Floating-Point Division and Square Root Unit,” in IEEE Trans. on Computers, vol. 69, no. 2, Feb. 2020, pp. 274–287
- NT05 : N. Takagi, S. Kadowaki, and K. Takagi, “A Hardware Algorithm for Integer Division,” in Proc. IEEE Int. Symp. on Computer Arithmetic, 2005, pp. 1–7
- AT03 : A. F. Tenca, A. Shantilal, and M. Sinky, “A Radix-4 On-line Division Design and Its Application to Networks of On-line Modules,” in Proceedings of SPIE, vol. 5205, 2003, pp. 529–540

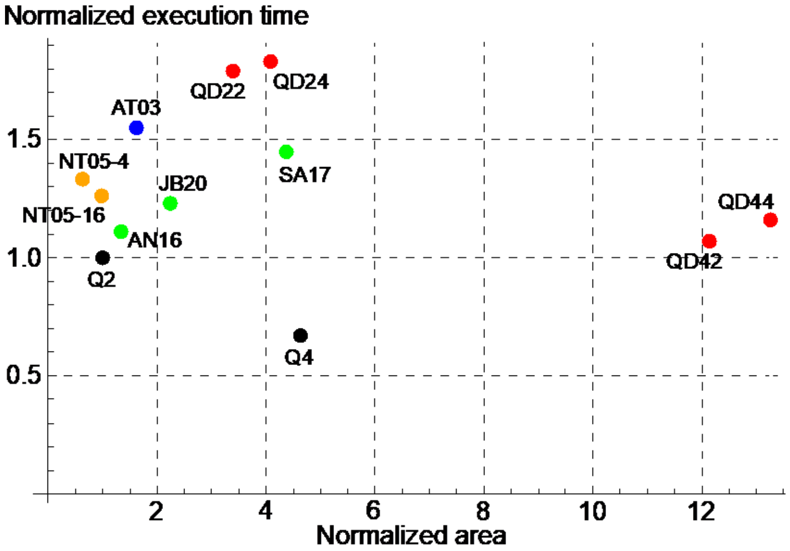
Result

- Scaled to Q2 value.
- Execution time = (clock period) * (#clock cycles)



	Offline dividers							Online dividers				
	AN16 [2]	SA17 [3]	JB20 [4]	NT05-4 [15]	NT05-16 [15]	Q2	Q4	AT03 [13]	QD22	QD24	QD42	QD44
Radix	8	16	64	4	16	4	16	4	4	4	16	16
Clock period (ns)	0.71	1.21	1.36	0.54	1.00	0.52	0.70	0.63	0.93	0.95	1.11	1.21
Execution time (ns)	18.46 (1.11)	24.20 (1.45)	20.40 (1.23)	22.14 (1.33)	21.00 (1.26)	16.64 (1.00)	11.20 **(0.67)**	25.83 (1.55)	29.76 (1.79)	30.40 (1.83)	17.76 (1.07)	19.36 (1.16)
Energy (pJ)	58.7 (2.47)	125.1 (5.27)	72.6 (3.06)	41.4 (1.74)	28.4 (1.19)	23.7 **(1.00)**	95.6 (4.03)	142.8 (6.02)	202.9 (8.55)	300.5 (12.66)	490.6 (20.67)	499.0 (21.02)
Area (um^2)	3,871 (1.34)	12,663 (4.37)	6,489 (2.24)	1,837 **(0.63)**	2,846 (0.98)	2,895 (1.00)	13,391 (4.63)	4,696 (1.62)	9,818 (3.39)	11,810 (4.08)	35,153 (12.14)	38,395 (13.26)

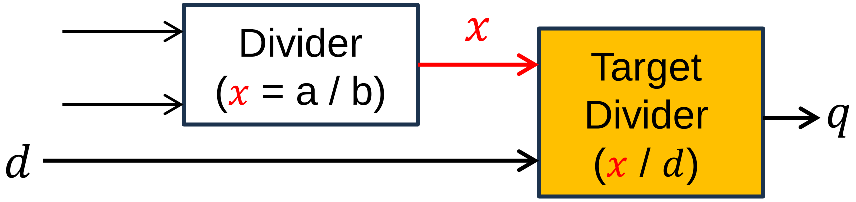
- In offline division :



Execution time simulation

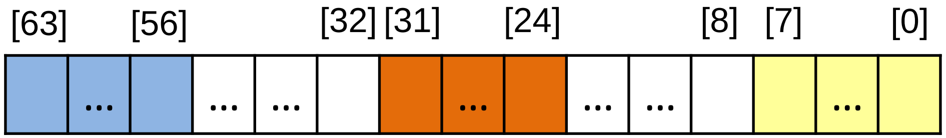
- # Simulation sets: 10,000

- Dependency type: Which operand is partially given (online)?
  - Example: Type-X



Dividend x	Divisor d	Dependency type
Online	Offline	Type-X
Offline	Online	Type-D
Online	Online	Type-XD

- Operand type: (the leading 1 location)
  - Large : idx ≥ [56]
  - Medium : [24] ≤ idx < [32]
  - Small : idx < [8]

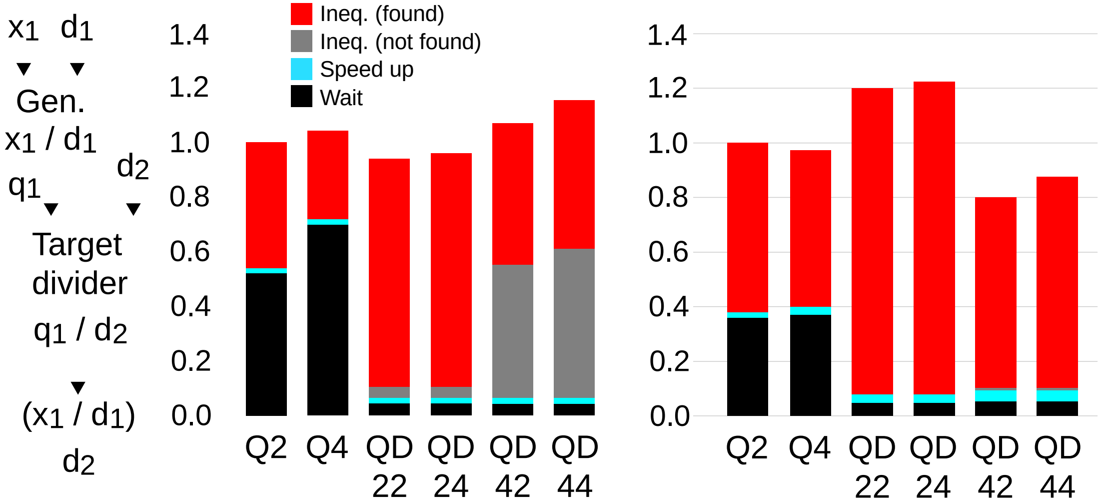


Computation Stages

- Stage
  1) Wait (⬛): Not enough # bits of the operands are given.
  2) Speed up (🟦): Leading 1's of both x and d are given (Path 1). This always takes 1 cycle.
  3) Ineq. (found) (🟥): We obtained a new digit of the quotient by the interval analysis in Path 2.
  4) Ineq. (not found) (▨): We could not obtain a new digit of the quotient by the interval analysis in Path 2.

Online Division: Type-X

- x: large, online
- d: small, offline
- Division: (large / small) / small = large / small

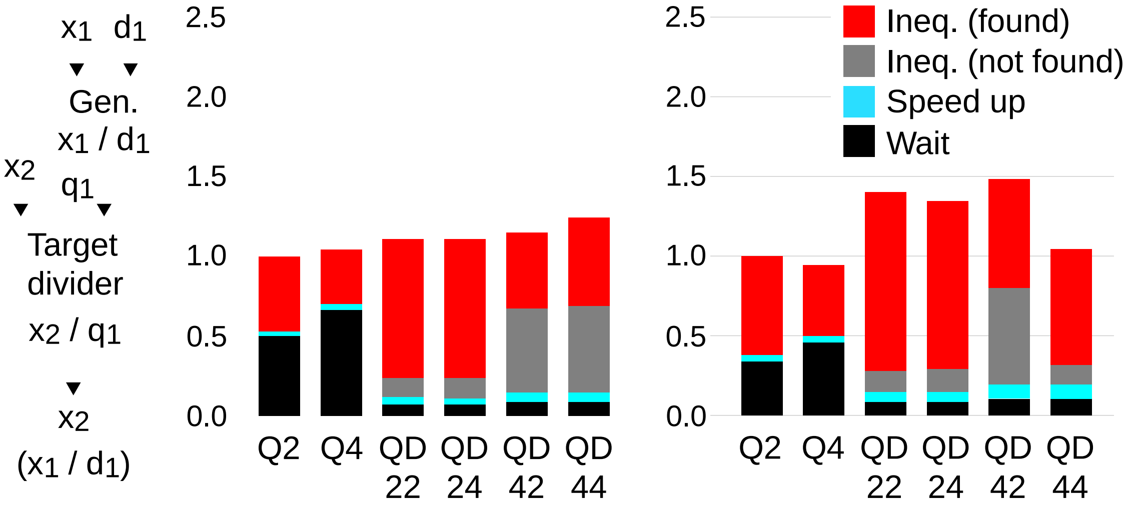


(a) (b) (c)

(Gen = 2bits / cycle) (Gen = 4bits / cycle)

Online Division: Type-D

- x: large, offline
- d: medium, online
- Division: large / (large / medium) = large / medium

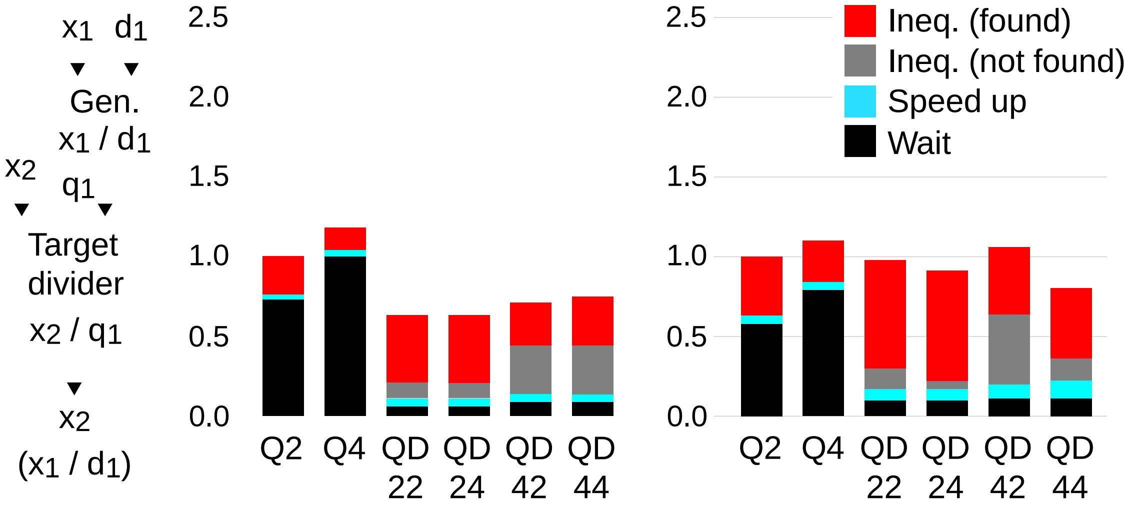


(a) (b) (c)

(Gen = 2bits / cycle) (Gen = 4bits / cycle)

Online Division: Type-D

- x: large, offline
- d: large, online
- Division: large / (large / small) = large / large



(a)

(b)

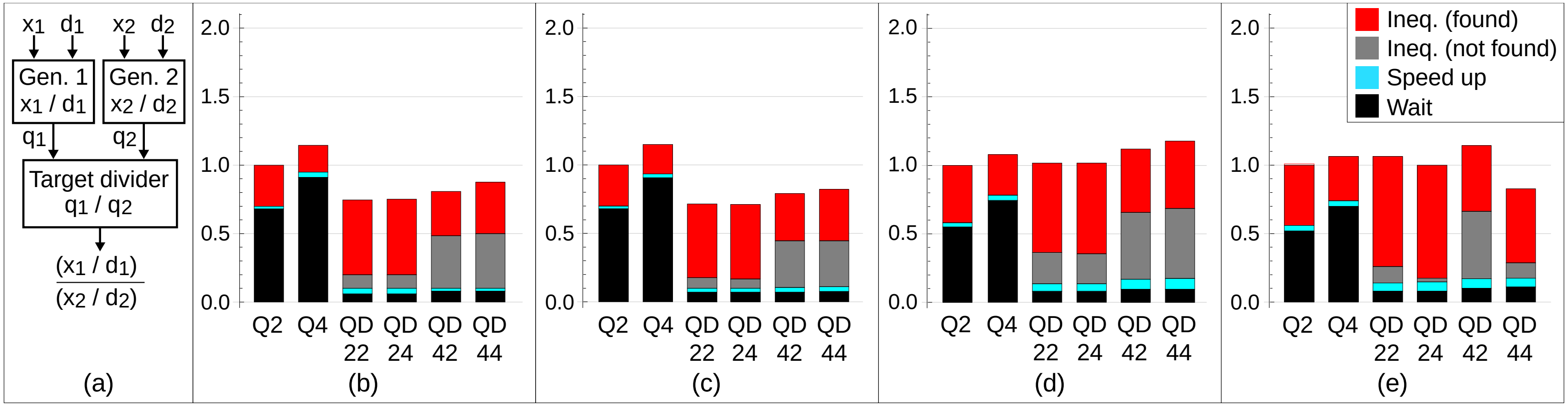
(Gen = 2bits / cycle)

(c)

(Gen = 4bits / cycle)

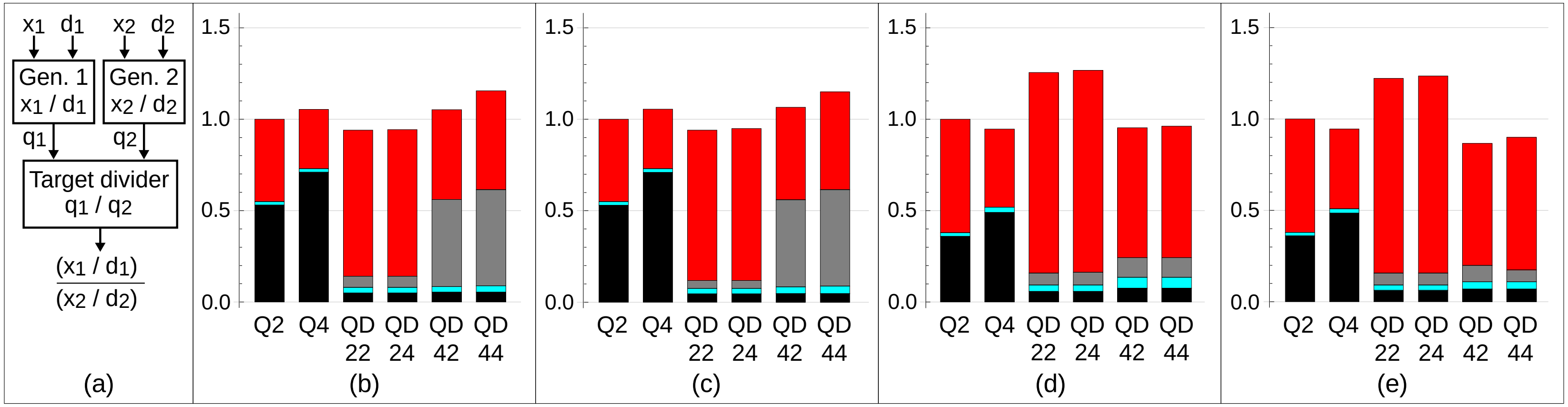
Online Division: Type-XD

- Division: (large / small) / (large / medium) = large / medium



(Gen1,Gen2) = (2bits, 2bits) (Gen1,Gen2) = (2bits, 4bits) (Gen1,Gen2) = (4bits, 2bits) (Gen1,Gen2) = (4bits, 4bits)

- Division: (large / small) / (large / large) = large / small



(Gen1,Gen2) = (2bits, 2bits) (Gen1,Gen2) = (2bits, 4bits) (Gen1,Gen2) = (4bits, 2bits) (Gen1,Gen2) = (4bits, 4bits)

Conclusion

- We proposed the interval-analysis-based division algorithm, which can be used for both offline and online division.
  - Radix-4
  - Radix-16
  - Uses the normal binary number system.

- Expected performance (for online division)
  - No single winner!
  - Depends on
    - Dependency types
    - Bit rate (# bits given per cycle)
    - Magnitudes of x and d.