

高基数 SRT 除算の論理回路実現に基づく回路構成と評価

葛 毅[†] 阿部 公輝[†] 浜田 穂積[†]

Abstract SRT 除算の基本的な構成は、商の桁の選択を論理回路で行う「論理回路実現」と表を引く事で行う「テーブル実現」に大きく分かれる。論理回路実現では、基数を r 、部分剰余を R 、除数を D 、各重複領域の境界線を kD とすると、 $rR - kD$ の符号をみることで商の桁 q を選択する。重複領域が存在するため $rR - kD$ の計算は、誤差を含んだ $\widehat{rR - kD}$ でよい。 k は一般的には生成しやすい重複領域の中心線 $k = q - 1/2$ をとる。本論文で検討する構成では桁上げ伝搬加算器 (CPA) のみが $\widehat{rR - kD}$ の速度を決定する。本論文では、 k を中心線以外にとることで、CPA を 1 ビット減らせることを示す。また、論理回路実現とテーブル実現を比較した結果を示す。同じ基数毎に比較した結果、基数 4、8、16 で各々、54 ビットでは速度が 3%、5%、6% 速くなり、面積が 1%、0%、19% 大きくなった。114 ビットでは速度が 4%、8%、9% 速くなり、面積が 1%、-3%、11% 大きくなった。

Hardware Organization and Evaluation of High-Radix SRT Division Based on the Logical Circuit Realization

Yi GE[†], Kôki ABE[†] and Hozumi HAMADA[†]

Abstract The hardware organization of the SRT division is categorized into two classes which we call the logical circuit realization and the table realization, where quotient digits are selected by means of logical circuits and a look-up table, respectively. In the logical circuit realization, quotient digits q are selected by examining the sign digit of $rR - kD$, where r is the radix, R is the partial remainder, D is the divider, and kD is a boundary line dividing the overlap region. Because of the overlap region, $\widehat{rR - kD}$ containing an error can be used in selection instead of the exact value of $rR - kD$. In convention the constant $k = q - 1/2$ is employed so that the line kD divides evenly the overlap region for ease of generating the value of k . In the organization of $\widehat{rR - kD}$ examined in this paper, a carry propagate adder (CPA) determines the delay of the critical path. This paper describes that we can reduce the CPA by one bit by employing other values for k than the conventional one. The results of comparing the logical circuit realization with the table realization are also given. In case of 54 bit divisor based on radices 4, 8, 16, speedups of 3%, 5%, 6% are obtained with 1%, 0%, 19% larger area costs than those of the table realization, respectively. In case of 114 bit divisor based on radices 4, 8, 16, speedups of 4%, 8%, 9% are obtained with 1%, -3%, 11% larger area costs than those of the table realization, respectively.

1 はじめに

減算シフト型除算法 [1] は、基数を r 、部分剰余を R 、商の桁を q 、除数を D とすると、 $R_j = rR_{j-1} - q_jD$ を繰り返すことで行う。 $1 \leq j, j$ は整数。 R_0 は被除数。 $1/2 \leq D < 1$ とする。SRT 除算 [2] は、減算シフト型除算法で商の桁集合を、 $\{-q_{max}, \dots, -2, -1, 0, 1, 2, \dots, q_{max}\}$, $(r-1)/2 \leq q_{max} \leq r-1$ としたものである。 $|R| \leq \rho D$, $\rho = q_{max}/(r-1)$ を満たすように商の桁 q を選択する。ある商の桁 q を選択できる領域は、 $|R| \leq \rho D \Leftrightarrow |rR - qD| \leq \rho D \Leftrightarrow (q - \rho)D \leq rR \leq (q + \rho)D$ である。商の桁集合の冗長性より、隣り合う商の桁 q と $q-1$ を選択できる重複領域が存在する。

SRT 除算の基本的な構成は、商の桁の選択を論理回路で行う「論理回路実現」[8] と表を引く事で行う「テーブル実現」[3][4] に大きく分かれる。論理回路実現では、各重複領域の境界線を kD とすると、 rR と kD の比較、つまり $rR - kD$ の符号をみることで q を選択する。重複領域が存在するため $rR - kD$ の計算は、誤差を含んだ $\widehat{rR - kD}$ でよい。 k は一般的には生成しやすい重複領域の中心線 $k = q - 1/2$ をとる [1][2][5][8]。本論文で検討する構成では、桁上げ伝搬加算器 (CPA) のみが $\widehat{rR - kD}$ の速度を

決定する。本論文では (1) k を一般的な中心線 $k = q - 1/2$ 以外にとることで、CPA のビット数を 1 ビット減らせることを示す。(2) 論理回路実現とテーブル実現を 54、114 ビットで設計し比較した結果を示す。2 章では論理回路実現の回路構成を示す。3、4 章で回路構成を評価する。5 章で論理回路実現とテーブル実現の比較結果を示す。

2 論理回路実現の回路構成

論理回路実現では、各重複領域の境界線を kD とすると、 $rR - kD$ の符号をみることで q を選択する。 $rR - kD$ の計算は、誤差を含んだ $\widehat{rR - kD}$ でよい。図 1 に $\widehat{rR - kD}$ の符号ビットを計算する回路構成を示す。 $\widehat{rR_c}$, $\widehat{rR_s}$ から $\widehat{rR - kD}$ がクリティカルパスになり、どの構成も CPA2 のみが $\widehat{rR - kD}$ の速度を決定する。CPA2 は出力が符号ビットのみの加算器であり、比較器とほぼ等しい速度である。 $\widehat{rR - kD}$ を行う回路は、部分剰余の表現が桁上げ保存形式か冗長 2 進表現かにより異なる。図 1(a) の桁上げ保存形式では、部分剰余の二つの値 $\widehat{rR_s}$, $\widehat{rR_c}$ と $-kD$ を桁上げ保存加算器 (CSA) CSA1 で \widehat{S} , \widehat{C} の 2 つに減らし CPA2 で加算する。 $\widehat{S} + \widehat{C}$ の誤差が $rR - kD$ の許容誤差範囲内であればよい。冗長 2 進表現では冗長 2 進加算器 (RBA) が 4 入力であるため、図 1(b),(c) の 2 種類の構成を考える。

図 2 に基数 4, $q_{max} = 2$ の論理回路実現による構成例を

[†]電気通信大学情報工学科。Department of Computer Science, The University of Electro-Communications.

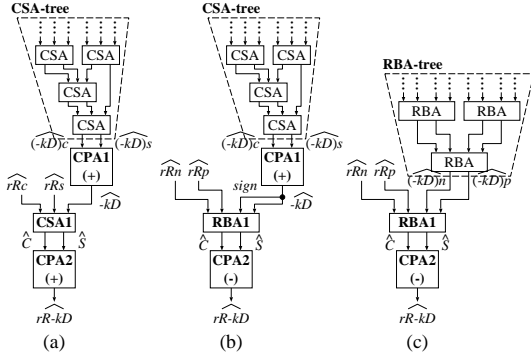


図 1: 部分剰余に桁上げ保存形式 (a) と冗長 2 進表現 (b), (c) を用いた場合の $rR - kD$ の構成

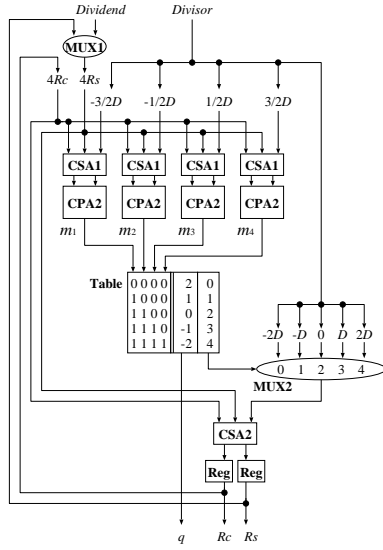


図 2: 基数 4, $q_{max} = 2$ の論理回路実現による構成例

示す。図 2 は桁上げ保存形式で $k = 3/2, 1/2, -1/2, -3/2$ とした場合である。CSA1、CPA2 で各 $rR - kD$ を計算し、中央の表 Table で各 $rR - kD$ の符号ビットから q を選択して、MUX2 の制御信号を作る。MUX2 で $-qD$ を選択し、CSA2 で $rR - qD$ を計算する。結果の R_s, R_c はレジスタ Reg に保持される。

3 $rR - kD$ の許容誤差

$rR - kD$ の許容誤差を計算する。ある重複領域の境界線を kD とすると、

$$(q - \rho) \leq k \leq (q - 1 + \rho) \quad (1)$$

である。 $rR - kD$ の誤差を持った値を $\widehat{rR - kD}$ 、誤差を Δ とすると、

$$\widehat{rR - kD} = (rR - kD) - \Delta \quad (2)$$

である。図 3 にある重複領域の P-D プロットを示す。 $rR - kD$ の意味を考えると、 rR が重複領域の上の境界 $(q - 1 + \rho)D$ より大きいとき、つまり、 $rR > (q - 1 + \rho)D$ の場合、 $\widehat{rR - kD} = rR - kD - \Delta \geq 0$ であるような

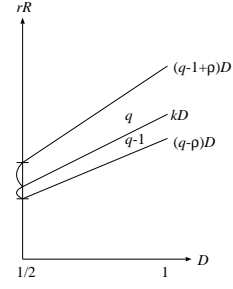


図 3: ある重複領域における P-D プロット

Δ が許容誤差である。このとき、 $rR - kD - \Delta$ の下限は $rR = (q - 1 + \rho)D, D = 1/2$ のときである。よって、 $rR - kD - \Delta > (q - 1 + \rho)/2 - k/2 - \Delta \geq 0 \Leftrightarrow ((q - 1 + \rho) - k)/2 \geq \Delta$ である。

同様に、 $rR < (q - \rho)D$ の場合、 $rR - kD - \Delta < 0$ であるような Δ が許容誤差である。このとき、 $rR - kD - \Delta$ の上限は $rR = (q - \rho)D, D = 1/2$ のときである。よって、 $rR - kD - \Delta < (q - \rho)/2 - k/2 - \Delta \leq 0 \Leftrightarrow ((q - \rho) - k)/2 \leq \Delta$ であればよい。よって、 $rR - kD$ の許容誤差は、

$$\frac{(q - \rho) - k}{2} \leq \Delta \leq \frac{(q - 1 + \rho) - k}{2} \quad (3)$$

である。

4 CPA2 のビット数

CPA2 の入力 S, C まだが完全に存在している $S + C = rR - kD$ の場合について、 k の条件とそのときの CPA2 の最小ビット数を小数ビット数 c と整数ビット数 i に分けて示す。それから、 $S + C \neq rR - kD$ の場合について示す。

4.1 小数ビット数 c

桁上げ保存形式の場合、CPA2 は加算である。よって、 $rR - kD = S + C$ を満たす二つの 2 の補数の値 S, C を考える。 S, C はそれぞれ CSA1 の出力のサム、キャリーとする。 S, C の誤差を含む値を \widehat{S}, \widehat{C} 、切捨て誤差を $\Delta S, \Delta C$ とすると、 $S = \widehat{S} + \Delta S, C = \widehat{C} + \Delta C$ である。ここで、 S, C はそれぞれ 2 の補数表現であるから、 $0 \leq \Delta S < 1/2^s, 0 \leq \Delta C < 1/2^c, s, c$ は整数。また、 $1/2^s \leq 1/2^c \Leftrightarrow c \leq s$ とする。

この場合は、 $rR - kD$ は小数点以下 s 桁までしかないため、許容誤差 Δ の上限をもう一度決める。許容誤差 Δ の上限は条件式 $\widehat{rR - kD} = rR - kD - \Delta \geq 0$ によって決まる。ここで、 $\widehat{rR - kD}$ は小数点以下 s 桁までしかないため条件式は $\widehat{rR - kD} = rR - kD - \Delta > -1/2^s$ とできる。よって Δ の上限は、 $\Delta < ((q - 1 + \rho) - k)/2 + 1/2^s$ である。よって、この場合 $rR - kD$ の許容誤差 Δ' は、

$$\frac{(q - \rho) - k}{2} \leq \Delta' < \frac{(q - 1 + \rho) - k}{2} + \frac{1}{2^s} \quad (4)$$

である。加算は小数点以下のビットの数が異なる場合、CPA のビット数は短い方の幅があればよい。よって、CPA2

表 1: 演算の種類と k の条件に対する、CPA2 の最小の小数ビット数 c 。 $f = \lceil \log_2(2/(2\rho - 1)) \rceil$ 。

type	operation of CPA2	condition of k	minimum value of c	condition of s
(I)	$\widehat{S} + \widehat{C}$	$(q - \rho) \leq k \leq (q - 1 + \rho) - 2^{-c+1}$	$c = f$	$s = \lceil -\log_2(\frac{(q-1+\rho)-k}{2} - \frac{1}{2^f}) \rceil$
(II)	$\widehat{S} + \widehat{C}$	$k = q - \frac{1}{2}$	$c = f + 1$	$s = \lceil -\log_2(\frac{2\rho-1}{4} - \frac{1}{2^{f+1}}) \rceil$
(III)	$\widehat{S} + \widehat{C} + \frac{1}{2^c}$	$(q - \rho) + 2^{-c+1} \leq k \leq (q - 1 + \rho)$	$c = f$	$s = \lceil -\log_2(\frac{(q-1+\rho)-k}{2}) \rceil$
(IV)	$\widehat{S} + \widehat{C} + \frac{1}{2^c}$	$k = q - \frac{1}{2}$	$c = f + 1$	$s = f + 1$

の小数ビット数は c である。また、CPA では加算と加算 +1 に対応する $rR - kD = \widehat{S} + \widehat{C}$, $\widehat{S} + \widehat{C} + 1/2^c$ が容易に行えるため、この 2 つの場合について考える。

4.1.1 CPA2 で $\widehat{S} + \widehat{C}$ を行う場合

$rR - kD = \widehat{S} + \widehat{C} = (S - \Delta S) + (C - \Delta C) = (S + C) - (\Delta S + \Delta C)$ である。ここで誤差の範囲は、

$$0 \leq \Delta S + \Delta C < \frac{1}{2^s} + \frac{1}{2^c} \quad (5)$$

である。式 (5) が式 (4) の範囲内であればよいから、

$$\frac{1}{2^c} \leq \frac{(q - 1 + \rho) - k}{2} \quad (6)$$

であればよい。ここで、 $D = 1/2$ における重複領域の最大幅から、 $1/2^f \leq (q - 1 + \rho)/2 - (q - \rho)/2$

$$\Leftrightarrow f \geq \log_2\left(\frac{2}{2\rho - 1}\right) \quad (7)$$

を満たす最小の整数 f を導入する。最小の c を得る条件と、 k が一般的な中心線 $k = q - 1/2$ の場合の c を求める。

最小の c は、式 (6) で k が最小の場合であるから、式 (6)(1)(7) より、 $1/2^c \leq ((q - 1 + \rho) - (q - \rho))/2 \Leftrightarrow c \geq \log_2(2/(2\rho - 1)) \Leftrightarrow c = f$ 。このときの k は式 (6)(1) より、 $1/2^c \leq ((q - 1 + \rho) - k)/2 \leq ((q - 1 + \rho) - (q - \rho))/2 \Leftrightarrow (q - \rho) \leq k \leq (q - 1 + \rho) - 2^{-c+1}$ 。 k が一般的な中心線 $k = q - 1/2$ の場合は、式 (6) に代入して、 $c = f + 1$ 。

4.1.2 CPA2 で $\widehat{S} + \widehat{C} + 1/2^c$ を行う場合

$rR - kD = \widehat{S} + \widehat{C} + 1/2^c = (S - \Delta S) + (C - \Delta C) + 1/2^c = (S + C) - (\Delta S + \Delta C - 1/2^c)$ である。ここで誤差の範囲は、

$$-\frac{1}{2^c} \leq \Delta S + \Delta C - \frac{1}{2^c} < \frac{1}{2^s} \quad (8)$$

である。式 (8) が式 (4) の範囲内であればよいから、

$$\frac{1}{2^c} \leq \frac{k - (q - \rho)}{2} \quad (9)$$

であればよい。最小の c は、式 (9) で k が最大の場合であるから、式 (9)(1)(7) より、 $1/2^c \leq ((q - 1 + \rho) - (q - \rho))/2 \Leftrightarrow c \geq \log_2(2/(2\rho - 1)) \Leftrightarrow c = f$ 。このときの k は式 (9)(1) より、 $1/2^c \leq (k - (q - \rho))/2 \leq ((q - 1 + \rho) - (q - \rho))/2 \Leftrightarrow (q - \rho) + 2^{-c+1} \leq k \leq (q - 1 + \rho)$ 。 k が一般的な中心線 $k = q - 1/2$ の場合は、式 (9) に代入して、 $c = f + 1$ 。

式 (6)、(9) より、 s は関係なくなり、 c のみで決まる事が分かる。 $c = s$ とする必要がない。表 1 にまとめる。

表 2: CPA2 の最小ビット数 $i + f$ 、表 1 の $c = f$ の場合。 $k = q - 1/2$ の場合は $i + f + 1$ 。

r	q_{max}	i	f	$i + f$	r	q_{max}	i	f	$i + f$
4	2	3	3	6	16	8	5	5	10
	3	4	1	5		9	5	4	9
8	4	4	4	8		10	5	3	8
	5	4	3	7		11	5	3	8
	6	4	2	6		12	5	2	7
	7	5	1	6		13	5	2	7
						14	5	2	7
						15	6	1	7

$f(1/2^f$ は $D = 1/2$ における重複領域の最大幅より小さい最大の値である) を基準とすると、 k を重複領域の上または下の境界線に近づけることにより、 $c = f$ とできるが、 k を一般的な重複領域の中心線 $k = q - 1/2$ とすると、加算 +1 を行なっても (IV)、 $c = f$ と出来ない。すなわち、 k を変化させて許容誤差 Δ の範囲を $+1/2^f$ ずらすこと (I) と、加算 +1 で発生誤差を $-1/2^f$ ずらすこと (IV) が等価でない。付録に (I) の場合に $c = f$ とできることと (IV) の場合に $c = f$ とできないことの証明をつける。部分剰余が冗長 2 進表現の場合は \widehat{C} をビット反転した $\widehat{C} = \overline{\widehat{C}}$ とした場合と同じである。

s は $rR - kD = S + C$ の場合は必要ないが、 $rR - kD \neq S + C$ の場合、つまり **CSA1**, **CPA1**, **CSA-tree** を小さくする為に必要となる。このためには、これまでの議論で、 Δ' (式 (4)) を Δ (式 (3)) に置き換えて求めればよい。上の各場合における s を求めた結果も表 1 に示す。 $k = q - 1/2$ の場合の (II)、(IV) を比較すると、(IV) の方が s が小さいため、(II) に比べて **CSA1**, **CPA1**, **CSA-tree** を小さくすることが出来る。(II) は有効ではない。(I)(III) は k の方向が逆だけでほとんど同じである。なお、 $c = s$ の条件から出発すると、(IV) の $k = q - 1/2$ が得られる。**CPA1**, **CSA1(RBA1)**, **CPA2** の小数ビット数は各々図 1(a) では $s, c + 1, c$ 、(b) では $s + 1, c + 2, c$ である。**CPA1** の整数ビット数は $\log_2(2 \cdot (q_{max} - 1 + \rho))$ より大きい最小の整数である。(c) では各 RBA のビット数は、**CPA2** から数えて RBA の段が上がる毎に c に 2 づつ足した値である。(I) の c, s, k は、例えば基数 4, $q_{max} = 2$ の場合は $c = 3, s = 6, k = 11/8, 3/8, -5/8, -13/8$ である。

4.2 整数ビット数 i

CPA2 の整数ビット数 i を説明する。傾き k の大きなものから $k_1, k_2, \dots, k_{2q_{max}}$ とし、それぞれの $rR - kD$ の符号ビットを $m_1, m_2, \dots, m_{2q_{max}}$ とする。 $(-q_{max} - \rho) - k \leq rR - kD \leq (q_{max} + \rho) - (-k)$ であるから、 $rR - kD$ の最小の整数ビット数は、 $i' = \lceil \log_2(2q_{max} + 2\rho + 2k) \rceil$ (k は $|k_1|$ と $|k_{2q_{max}}|$ の大きい方) であるが、 $i' - 1$ とし

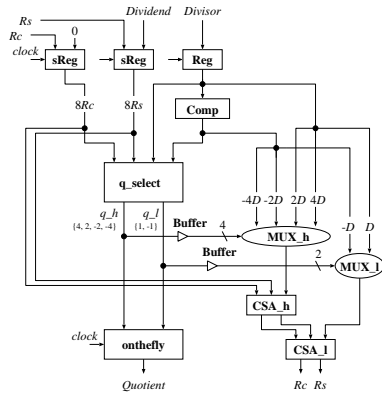


図 4: r8q4_logic の構成

表 3: 各実装の全体時間と面積 (54 ビット)

implementation	number of cycles	cycle time (ns)	total time (ns)	total area (μm^2)
r2	54	1.96	105.84	739980
r4q2_logic	28	2.64	73.92	892152
r4q2_table	27	2.82	76.14	880165
r8q4_logic	19	3.13	59.47	1219594
r8q7_table	18	3.46	62.28	1214911
r16q8_logic	15	3.25	48.75	1794623
r4x4q2_table	14	3.68	51.52	1503299
r32q16_logic	12	3.57	42.84	3388018

でも各 q を区別することができる。Table は整数ビット数を i' ビットとして正確に計算する場合、 $\pm q_{max}$ の場合は m_1 または $m_2 q_{max}$ の 1 ビット、その他の場合は m_1 と m_2 等の 2 ビットを見れば区別できる。よって、符号ビットが定まってから MUX2 の出力までの論理は、 $-qD = \overline{m_1} \cdot (-q_{max}D) + m_1 \cdot \overline{m_2} \cdot (-q_{max}-1)D + \dots + m_2 q_{max} - 1 \cdot \overline{m_2 q_{max}} \cdot (-(-q_{max}+1))D + m_2 q_{max} \cdot (-(-q_{max})D)$ である。よって、Table の速度は、MUX2 をデコードした値で制御すれば、 $m_1 \cdot \overline{m_2}$ 等の隣り合った符号ビットの 2 入力論理積 (2AND) であり論理 1 段である。一方、整数ビット数を $i' - 1$ ビットとする場合は、 k が大きいところと小さいところで m が反転するが、 k が大きいところでは m_1 から連続して 0、 k が小さいところでは $m_2 q_{max}$ から連続して 1 となるため、 q_{max} の場合には m_1 と $m_2 \cdot (q_{max} - 2^{\lfloor \log_2 q_{max} \rfloor + 1})$ の 2 ビット、 $-q_{max}$ の場合には $m_2 \cdot (2^{\lfloor \log_2 q_{max} \rfloor + 1}) + 1$ と $m_2 q_{max}$ の 2 ビット、その他の場合は m_1 と m_2 等の 2 ビットを見れば区別できる。よって、Table の速度は 2AND のままで整数ビット数を 1 ビット減らせる。よって、CPA2 の整数ビット数 i は、 $q_{max} = r - 1$ の場合を除き $i = i' - 1$ 、 $q_{max} = r - 1$ の場合は q_{max} と $-q_{max}$ が区別できなくなるため、 $i = i'$ である。各 q に対する加算の種類と k の条件 (表 1) は独立であるから、 $rR - kD$ の整数ビット数は、 $i = i' - 1$ となるように k を選択できる。表 2 に CPA2 の最小ビット数の計算結果を示す。

5 論理回路実現とテーブル実現の比較結果

テーブル実現と論理回路実現で各々数種類の回路 [5] を Verilog-HDL で設計し、論理合成ツール DesignCompiler で合成して得られる速度と面積の値を用いて比較した。

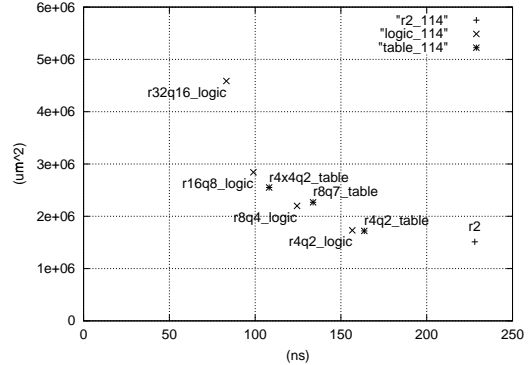
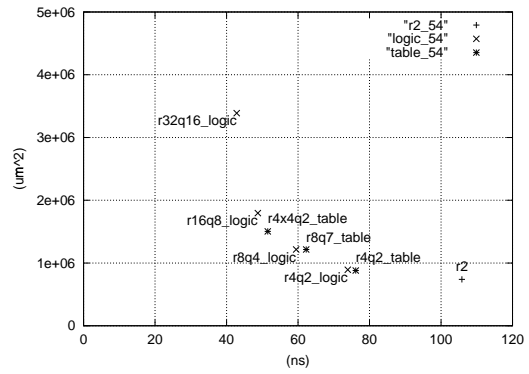


図 5: 各実装の比較、54 ビット (上) と 114 ビット (下)

ゲート間の駆動遅延は含む。配線遅延は含まない。合成に用いたセルライブラリはローム株式会社の製造条件に基づき VDEC (東京大学大規模集積システム設計教育研究センター) で製作されたものを用いた。主な製造条件は、CMOS 0.6 μm 、PolySi2 層、メタル配線 3 層である。

設計した回路は次の通り。54、114 ビットである。丸めは実装していない。論理回路実現は表 1 (IV) である。

- r2 : 基数 2 の SRT 除算。
- r4q2_logic : 基数 4, $q_{max} = 2$ の論理回路実現。
- r4q2_table : 基数 4, $q_{max} = 2$ のテーブル実現。
- r8q4_logic : 基数 8, $q_{max} = 4$ の論理回路実現。
- r8q7_table : 基数 8, $q_{max} = 7$ のテーブル実現。
- r16q8_logic : 基数 16, $q_{max} = 8$ の論理回路実現。
- r4x4q2_table : 基数 4, $q_{max} = 2$ のテーブル実現を二つオーバーラップさせた基数 16 の構成 [7]。
- r32q16_logic : 基数 32, $q_{max} = 16$ の論理回路実現。

図 4 に r8q4_logic の構成を示す。図 4 は図 2 にほぼ対応している。基数 2, 4 の構成では qD を選択する MUX と CSA は 1 段であるが、基数 8 以上では、2 段 (基数 8, 16) または 3 段 (基数 32) 構成としている部分が異なる。論理回路実現もテーブル実現も q_select を除き、全体の構成はほぼ同じである。sReg、Reg はレジスタ、Comp は 2 の補数を取る回路、q_select は商の桁を選択する回路、Buffer はバッファ、MUX はデコードされた信号 q_h, q_l で制御するマルチプレクサ、onthefly は商の桁 q から商を得るための on the fly 変換 [6] を行なう回路である。

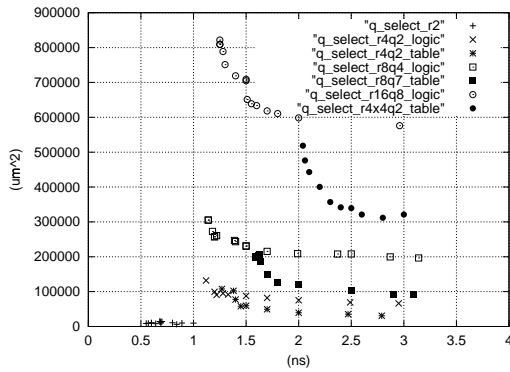


図 6: q_select の比較

表 3、図 5 は 54、114 ビットの各実装の全体時間と面積の比較結果である。 q_select は階層を破壊して合成しており、冗長な論理は取り除かれている。テーブル実現の表は case 文で記述し、論理を圧縮して合成した回路になっている。論理回路実現では kD を作っておく必要があるため、サイクル数を 1 サイクル多く数える。同じ基数毎に比較すると、基数 4、8、16 で各々、54 ビットでは、速度が 3%、5%、6% 速くなり、面積が 1%、0%、19% 大きくなった。114 ビットでは、速度が 4%、8%、9% 速くなり、面積が 1%、-3%、11% 大きくなった。

図 6 は各実装で主に異なる、モジュール q_select を遅延時間の制約条件を変化させて合成した結果である。時間は繰り返しのクリティカルパスである。モジュール間の駆動遅延は含まない。論理回路実現はテーブル実現に比べて同じ基数では速度は速い。 q_select では、実質的にどちらの実現法でも処理は同じであるが、論理回路実現ではその一部分である kD を作る部分を前処理としているためである。面積では大きくなっている。しかし、基数が小さいと q_select の面積の全体の中の割合が小さいため、全体の面積の差はほとんど出ない(図 5、基数 4、8)。基数が大きくなると全体の中の割合が大きくなり、全体の面積の差が出てくる(基数 16)。表 1(1) とすると (IV) とした場合に比べて、基数 4 では q_select の面積は約 1.5 倍、全体の面積は 5% 大きい。表 4 は各実装のサイクル時間(クリティカルパス)と面積の内訳である。各実装で q_select 以外は基本的に同じモジュールは同じ構成要素を用いている。

6 おわりに

本論文では論理回路実現の回路構成を詳細に検討し、その結果、重複領域の境界線を一般的な中心線 $k = q - 1/2$ ではなく、重複領域の上または下の境界線に近づけることにより、CPA2 のビット数を 1 ビット小さくできることを示した。これは、傾き k を変化させて許容誤差 Δ の範囲を $+1/2^f$ ずらすことと、加算 +1 により発生誤差を $-1/2^f$ ずらすことが等価にならないためである。

論理回路実現とテーブル実現の 54、114 ビットの回路を各々設計し比較した。主な違いはモジュール q_select である。論理回路実現では kD を作る部分を前処理として

表 4: 各実装のサイクル時間と面積 (54 ビット)

module	r4q2_logic		r4q2_table	
	time (ns)	area (μm^2)	time (ns)	area (μm^2)
sReg	0.54	175033	0.49	175033
Reg	-	64298	-	64298
Comp	-	53899	-	53899
q_select	1.30	97082	1.52	85095
Buffer	0.14	22861	0.14	22861
MUX	0.31	124309	0.31	124309
CSA	0.35	66044	0.35	66044
onthe-fly	-	288626	-	288626
total	2.64	892152	2.82	880165

module	r8q4_logic		r8q7_table	
	time (ns)	area (μm^2)	time (ns)	area (μm^2)
sReg	0.51	176700	0.50	178367
Reg	-	64298	-	64298
Comp	-	53899	-	53899
q_select	1.43	278941	1.77	191306
Buffer	0.14	34292	0.14	45723
MUX_h	0.31	122007	0.31	122007
MUX_l	-	64298	-	126611
CSA_h	0.39	64774	0.39	66044
CSA_l	0.35	67314	0.35	67314
onthe-fly	-	290769	-	299342
total	3.13	1215546	3.46	1214911

module	r16q8_logic		r4x4q2_table	
	time (ns)	area (μm^2)	time (ns)	area (μm^2)
sReg	0.55	175033	0.52	175033
Reg	-	64298	-	64298
Comp	-	53899	-	53899
q_select	1.51	774749	2.36	460642
Buffer	0.14	45723	0.14	45723
MUX_h	0.31	119705	-	124309
MUX_l	-	124309	0.31	124309
CSA_h	0.39	63504	-	66044
CSA_l	0.35	66044	0.35	66044
onthe-fly	-	307359	-	322997
total	3.25	1794623	3.68	1503299

いるため毎回の繰り返しが速くなる。

謝辞 原稿を読んで頂き、貴重な助言を頂いた名古屋大学 高木直史教授、および鶴田三敏氏、SRT 除算について議論した佐藤竜一氏、貴重な指摘をして頂いた情報処理学会論文査読委員の方々、各種 CAD ツール、セルライブラリを提供して頂いた VDEC、およびローム株式会社に感謝致します。

参考文献

- [1] 高木直史: “除算回路のアルゴリズム,” 情報処理, vol. 37, no. 3, pp. 280-286, Mar. 1996.
- [2] J. E. Robertson: “A New Class of Digital Division Methods,” *IRE Trans. Electronic Computers*, vol. EC-7, no 9, pp. 218-222, Sept. 1958.
- [3] D. E. Atkins: “Higher-Radix Division Using Estimates of the Divisor and Partial Remainders,” *IEEE Trans. Computers*, vol. C-17, no 10, pp. 925-934, Oct. 1968.
- [4] N. Burgess and T. Williams: “Choices of Operand Truncation in the SRT Division Algorithm,” *IEEE Trans. Computers*, vol. 44, no. 7, pp. 933-938, July. 1995.
- [5] M. D. Ercegovac and T. Lang: “Division and Square Root – Digit-Recurrence Algorithms and Implementations,” Kluwer Academic Publishers, 1994.
- [6] M. D. Ercegovac and T. Lang: “On-the-Fly Conversion of Redundant into Conventional Representations,” *IEEE Trans. Computers*, vol. C-36, no. 7, pp. 895-897, July. 1987.
- [7] G. S. Taylor: “Radix 16 SRT Dividers With Overlapped Quotient Selection Stages,” *Proc. 7th IEEE Symp. Computer Arithmetic*, pp. 64-71, 1985.
- [8] 葛 毅, 阿部公輝: “高基数 SRT 除算の算術モデルに基づく回路構成と評価,” 情報処理学会第 62 回全国大会論文集, pp1-97-1-98, Mar. 2001.

A 付録：表 1(I)、(IV) の証明

k を重複領域の下の境界線に近づける (I) と $c = f$ とできることと、 k を重複領域の中心線 $k = q - 1/2$ とすると (IV) c の最小値を $c = f$ とできないことを証明する。 $1/2^f$ は $D = 1/2$ における重複領域の最大幅より小さい最大の値である。

許容誤差 Δ の最大値を $\Delta Max = (q - 1 + \rho - k)/2$ 、最小値を $\Delta Min = (q - \rho - k)/2$ とする。 $\Delta Min \leq \Delta \leq \Delta Max$ 、 $\Delta Max - \Delta Min = \rho - 1/2$ である。2 の補数の値 S, C の切捨て誤差を $\Delta S, \Delta C$ 、 $0 \leq \Delta S \leq 1/2^s$ 、 $0 \leq \Delta C \leq 1/2^c$ 、 $c \leq s$ とする。 s, c は整数。

A.1 (I) の証明

加算で $\widehat{S} + \widehat{C}$ を行なう場合は、 $S + C$ の誤差は $\Delta S + \Delta C$ である。ある s, c で

$$S + C - \Delta Max \geq 0 \Rightarrow S + C - (\Delta S + \Delta C) \geq 0 \quad (10)$$

$$S + C - \Delta Min < 0 \Rightarrow S + C - (\Delta S + \Delta C) < 0 \quad (11)$$

が成り立てば、 S, C を $s+1, c+1$ ビット以下切捨てて加算した $\widehat{S} + \widehat{C}$ の符号が、 $S + C$ に許容誤差 Δ の範囲の最大、最小値が発生した場合の値 $S + C - \Delta Max, S + C - \Delta Min$ の符号の範囲に収まる。よって、その s, c で商の桁を正しく選択できる。

定理 1 許容誤差 Δ が $-\alpha \leq \Delta \leq 1/2^f + \beta$ 、 $\Delta Min = -\alpha$ 、 $\Delta Max = 1/2^f + \beta$ 、 $f = \lceil \log_2(2/(2\rho - 1)) \rceil$ 、 $\alpha + \beta = (\rho - 1/2) - 1/2^f$ 、 $0 \leq \alpha < 1/2^f$ 、 $0 \leq \beta < 1/2^f$ のとき ((I) の条件)、 $c = s = f$ とすると、式 (10)(11) が成り立つ。

定理 1 の証明 S, C の小数点 $s+1, c+1$ ビット以下を切捨てる時に加算 $\widehat{S} + \widehat{C}$ で発生する切捨て誤差 $\Delta S + \Delta C$ は $((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s$ または $1/2^c + ((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s$ である ($S+C$ が $1/2^s$ の倍数よりも小さい最大の値の場合は、 $\Delta S + \Delta C = 1/2^c - 1/2^\infty$)。ここで $c = s$ の場合は、 $\Delta S + \Delta C = ((S+C) \cdot 2^c - \lfloor (S+C) \cdot 2^c \rfloor) / 2^c$ または $1/2^c + ((S+C) \cdot 2^c - \lfloor (S+C) \cdot 2^c \rfloor) / 2^c$ である。

2 の補数の値の符号は、その値を $1/2^c$ の倍数まで小さくしたものと等しい。よって、 $S + C - \Delta Max \geq 0 \Leftrightarrow S + C - 1/2^c - \beta \geq 0 \Leftrightarrow S + C - 1/2^c \geq 0 \Leftrightarrow \lfloor (S+C - 1/2^c) \cdot 2^c \rfloor / 2^c \geq 0 \Leftrightarrow S + C - 1/2^c - ((S+C - 1/2^c) \cdot 2^c - \lfloor (S+C - 1/2^c) \cdot 2^c \rfloor) / 2^c \geq 0 \Leftrightarrow S + C - (1/2^c + ((S+C) \cdot 2^c - \lfloor (S+C) \cdot 2^c \rfloor) / 2^c) \geq 0 \Leftrightarrow S + C - (\Delta S + \Delta C) \geq 0$ 。よって、式 (10) が成り立つ。

同様に、 $S + C - \Delta Min < 0 \Leftrightarrow S + C + \alpha < 0 \Leftrightarrow S + C < 0 \Leftrightarrow \lfloor (S+C) \cdot 2^c \rfloor / 2^c < 0 \Leftrightarrow S + C - (\Delta S + \Delta C) < 0$ 。よって、式 (11) が成り立つ。(証明終)

定理 1 より CPA2 で $\widehat{S} + \widehat{C}$ 、 k を $(q - \rho) \leq k \leq (q - 1 + \rho) - 2^{-c+1}$ とすれば (I)、 $c = f$ とできる。

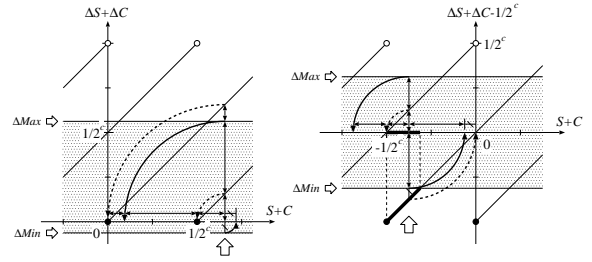


図 7: 定理 1(左)、2(右) の図 ($c = s = f$)。

A.2 (IV) の証明

加算で $\widehat{S} + \widehat{C} + 1/2^c$ を行なう場合は、 $S + C$ の誤差は $\Delta S + \Delta C - 1/2^c$ である。ある s, c で

$$S + C - \Delta Max \geq 0 \Rightarrow S + C - (\Delta S + \Delta C - 1/2^c) \geq 0 \quad (12)$$

$$S + C - \Delta Min < 0 \Rightarrow S + C - (\Delta S + \Delta C - 1/2^c) < 0 \quad (13)$$

が成り立てば、その s, c で商の桁を正しく選択できる。

定理 2 許容誤差 Δ が $-1/2^{f+1} - \alpha \leq \Delta \leq 1/2^{f+1} + \beta$ 、 $\Delta Min = -1/2^{f+1} - \alpha$ 、 $\Delta Max = 1/2^{f+1} + \beta$ 、 $f = \lceil \log_2(2/(2\rho - 1)) \rceil$ 、 $\alpha + \beta = ((\rho - 1/2) - 1/2^f) / 2$ 、 $0 \leq \alpha < 1/2^{f+1}$ 、 $0 \leq \beta < 1/2^{f+1}$ のとき ((IV) の条件)、 $c = f$ 、 $c \leq s$ とすると、式 (13) は成り立たない ($c = s = f$ の場合も式 (13) は成り立たない)。

定理 2 の証明 $c \leq s$ の場合、 $\widehat{S} + \widehat{C} + 1/2^c$ で発生する切捨て誤差 $\Delta S + \Delta C - 1/2^c$ は $-1/2^c + ((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s$ または $((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s$ である ($S + C$ が $1/2^s$ の倍数よりも小さい最大の値の場合は、 $\Delta S + \Delta C - 1/2^c = -1/2^\infty$)。

例えば、 $S + C = -1/2^c$ 、 $\Delta S + \Delta C - 1/2^c = -1/2^c + ((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s$ とすると、 $S + C - \Delta Min = -1/2^c + 1/2^{c+1} + \alpha < 0$ 。また、 $S + C - (\Delta S + \Delta C - 1/2^c) = S + C - (-1/2^c + ((S+C) \cdot 2^s - \lfloor (S+C) \cdot 2^s \rfloor) / 2^s) = -1/2^c - \lfloor (-1/2^c) \cdot 2^s \rfloor / 2^s = 0$ 。よって、式 (13) は成り立たない。(証明終)

定理 2 より $\widehat{S} + \widehat{C} + 1/2^c$ 、一般的な $k = q - 1/2$ の場合 (IV) は $c = f$ とできない。

図 7 に定理 1、2 の証明に対応する図を示す。図は横軸が $rR - kD = S + C$ の値、縦軸が $c = s = f$ の場合に $\widehat{S} + \widehat{C}$ 、 $\widehat{S} + \widehat{C} + 1/2^c$ で発生する誤差 $\Delta S + \Delta C$ 、 $\Delta S + \Delta C - 1/2^c$ (斜線) である。また、同時に許容誤差 $\Delta Min \leq \Delta \leq \Delta Max$ (網目部) を描いたものである。図左 ((I) の場合) では $c = s = f$ とできるが、図右 ((IV) の場合) では $c = s = f$ とすると、 $rR - kD = S + C$ が太線の範囲の値の場合には、 $\widehat{S} + \widehat{C} + 1/2^c$ は負でなければならないが、発生誤差が $-1/2^c + ((S+C) \cdot 2^c - \lfloor (S+C) \cdot 2^c \rfloor) / 2^c$ の場合 (斜めの太線) に $\widehat{S} + \widehat{C} + 1/2^c = 0$ となる。よって、式 (13) は成り立たず、 $c = s = f$ とできない。