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[†]

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」と表を引くことで行う「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．これまで，論理回路実現（$rR-kD$ の符号により商の桁を選択する方法）については $k=q-1/2$ の場合を含めて詳細に分析されておらず，その有効性は示されていない．本論文では論理回路実現に基づいて高速になる回路構成を提案し，その回路構成をすべての基数と商の最大値 q_{max} および k の条件について詳細に分析し，提案手法の有効性を示す．提案した回路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，提案した回路構成の有効性を示す．同じ基数ごとに比較した結果，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[†]

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. So far the logical circuit realization has not been examined in detail including the case of $k=q-1/2$, and its effectiveness has not been shown. In this paper, we propose a hardware organization of the logical circuit realization which results in a high performance divider. We also analyze in detail the organization with possible radices, maximum quotient digits q_{max}, and conditions of k. In the proposed organization 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. It also shows the effectiveness of the proposed hardware organization by comparing the logical circuit realization with the table realization. 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．$1 \le j$，j は整数．R_0 は被除数．$1/2 \le D < 1$ とする．SRT 除算[2)]は，減算シフト型除算法で商の桁集合を，$\{-q_{max},\ldots,-2,-1,0,1,2,\ldots,q_{max}\}$，$(r-1)/2 \le q_{max} \le r-1$ としたものである．$|R| \le \rho D$，$\rho = q_{max}/(r-1)$ 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，$|R| \le \rho D \Leftrightarrow |rR-qD| \le \rho D \Leftrightarrow (q-\rho)D \le rR \le (q+\rho)D$ である．商の桁集合の冗長性より，隣り合う商の桁 q と

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

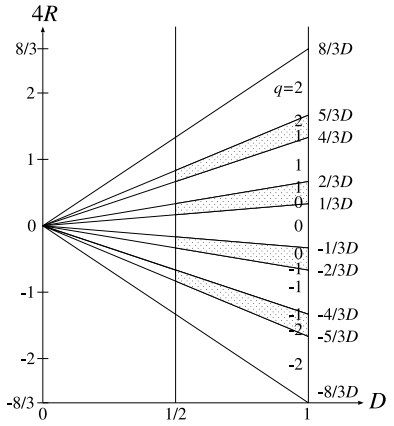

図 1 基数 4，$q_{max} = 2$ の P-D プロット
Fig. 1 The P-D plot in case of $r = 4$ and $q_{max} = 2$.

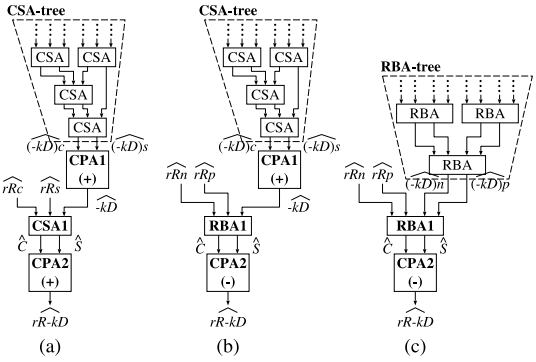

図 2 部分剰余に桁上げ保存形式 (a) と冗長 2 進表現 (b)，(c) を用いた場合の $r\widehat{R-k}D$ の構成
Fig. 2 Hardware organizations of $r\widehat{R-k}D$ using carry-save form (a), and redundant-binary forms (b) and (c).

$q-1$ を選択できる重複領域が存在する．

たとえば，基数 4，商の桁の最大値 $q_{max} = 2$ では，商の桁集合が $\{-2,-1,0,1,2\}$，$\rho = 2/3$ である．よって，ある商の桁 q を選択できる領域は，$(q-2/3)D \leq 4R \leq (q+2/3)D$ である．**図 1** に P-D プロットを示す．網目の部分が重複領域である．たとえば，商の桁 $q = 1$ と $q = 2$ を選択できる領域は，各々，$1/3D \leq 4R \leq 5/3D$，$4/3D \leq 4R \leq 8/3D$ であり，$4/3D \leq 4R \leq 5/3D$ の領域は $q = 1$，2 のどちらを選択してもよい．

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」[3]と表を引くことで行う「テーブル実現」[4),5)]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 $r\widehat{R-k}D$ 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)~3),6)]．

これまで，論理回路実現（$rR-kD$ の符号により商の桁を選択する方法）については $k = q-1/2$ の場合を含めて詳細に分析されておらず，その有効性は示されていない．本論文では論理回路実現に基づいて高速になる回路構成を提案し，その回路構成をすべての基数と q_{max} および k の条件について詳細に分析し，提案手法の有効性を示す．提案した回路構成では，桁上げ伝搬加算器（CPA）のみが $r\widehat{R-k}D$ の速度を決定する．本論文では，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．また，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，提案した回路構成の有効性を示す．

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 $r\widehat{R-k}D$ でよい．**図 2** に $r\widehat{R-k}D$ 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 $r\widehat{R-k}D$ がクリティカルパスになり，どの構成も **CPA2** のみが $r\widehat{R-k}D$ 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．$r\widehat{R-k}D$ 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 2 (a) の桁上げ保存形式では，部分剰余の 2 つの値 $\widehat{rR_s}$，$\widehat{rR_c}$ と $\widehat{-kD}$ 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 2 (b), (c) の 2 種類の構成を考える．

図 3 に基数 4，$q_{max} = 2$ の論理回路実現による構成例を示す．図 3 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 $r\widehat{R-k}D$ を計算し，中央の表 **Table** で各 $r\widehat{R-k}D$ の符号ビットから q を選択して，マルチプレクサ **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．

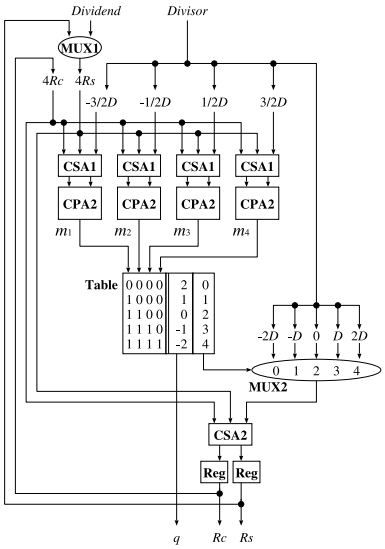

図 3 基数 4，$q_{max} = 2$ の論理回路実現による構成例
Fig. 3 A hardware organization based on the logical circuit realization in case of $r = 4$ and $q_{max} = 2$.

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 $r\widehat{R-k}D$，誤差を Δ とすると，

$$r\widehat{R-k}D = (rR - kD) - \Delta \quad (2)$$

である．**図 4** 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，$r\widehat{R-k}D = rR - kD - \Delta \geq 0$ であるような 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．

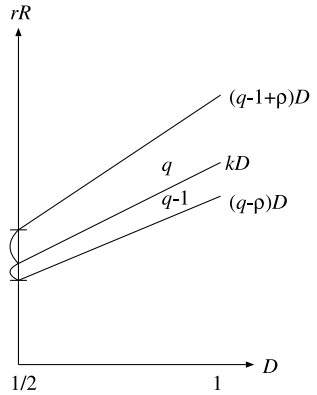

図 4 ある重複領域における P-D プロット
Fig. 4 An overlap region in the P-D plot.

4. CPA2 のビット数

まず，図 2 (a) において **CSA1**，**CPA1** がフルビットの場合について，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．次に，**CSA1**，**CPA1** がフルビットでない場合を考えて，許容誤差に収まるための **CSA1**，**CPA1** の最小ビット数について示す．

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 2 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 ΔS，ΔC 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 $r\widehat{R-k}D = rR - kD - \Delta \geq 0$ によって決まる．ここで，$r\widehat{R-k}D$ は小数点以下 s 桁までしかないため条件式は $r\widehat{R-k}D = rR - kD - \Delta > -1/2^s$ 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** 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 $r\widehat{R-k}D = \widehat{S} + \widehat{C}$，$\widehat{S} + \widehat{C} + 1/2^c$ が容易に行えるため，この 2 つの場合

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．$f = \lceil \log_2(2/(2\rho-1)) \rceil$
Table 1 Minimum of c (fraction bits of **CPA2**) for operations of **CPA2** and conditions of k. $f = \lceil \log_2(2/(2\rho-1)) \rceil$.

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

について考える．

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

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

$$0 \le \Delta S + \Delta C < \frac{1}{2^s} + \frac{1}{2^c} \tag{5}$$

である．式 (5) が式 (4) の範囲内であればよいから，

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

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，$1/2^f \le (q-1+\rho)/2-(q-\rho)/2$

$$\Leftrightarrow \underline{f \ge \log_2\left(\frac{2}{2\rho-1}\right)} \tag{7}$$

を満たす最小の整数 f を導入する．最小の c を得る条件（I）と，k が一般的な中心線 $k=q-1/2$ の場合の c（II）を各々求める．

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，$1/2^c \le ((q-1+\rho)-(q-\rho))/2 \Leftrightarrow c \ge \log_2(2/(2\rho-1)) \Leftrightarrow \underline{c=f}$．このときの k は式 (6)，(1) より，$1/2^c \le ((q-1+\rho)-k)/2 \le ((q-1+\rho)-(q-\rho))/2 \Leftrightarrow \underline{(q-\rho) \le k \le (q-1+\rho)-2^{-c+1}}$．$k$ が一般的な中心線 $k=q-1/2$ の場合は，式 (6) に代入して，$\underline{c=f+1}$．

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

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

$$-\frac{1}{2^c} \le \Delta S + \Delta C - \frac{1}{2^c} < \frac{1}{2^s} \tag{8}$$

である．式 (8) が式 (4) の範囲内であればよいから，

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

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，$1/2^c \le ((q-1+\rho)-(q-\rho))/2 \Leftrightarrow c \ge \log_2(2/(2\rho-1)) \Leftrightarrow \underline{c=f}$．このときの k は式 (9)，(1) より，$1/2^c \le (k-(q-\rho))/2 \le ((q-1+\rho)-(q-\rho))/2 \Leftrightarrow \underline{(q-\rho)+2^{-c+1} \le k \le (q-1+\rho)}$．$k$ が一般的な中心線 $k=q-1/2$ の場合は，式 (9) に代入して，$\underline{c=f+1}$．

式 (6)，(9) より，s は関係なくなり，c のみで決まることが分かる．s は $c \le s$ を満たせば何でもよく，$c=s$ とする必要がない．**表 1** にまとめる．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 を行っても（IV），$c=f$ とできない．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．

ここまでは，図 2 (a) の **CSA1**，**CPA1** がフルビットの場合について考えてきた．次に **CSA1**，**CPA1** がフルビットでない場合を考える．まず，**CSA1** の小数ビット数は **CPA2** より 1 ビット大きければ誤差は発生しない．次に，**CPA1** の小数ビット数を考える．これは，これまでの議論で，Δ'（式 (4)）を Δ（式 (3)）に置き換えて求めた s と等価である．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できる．（II）は有効ではない．（I），（III）は k の方向が逆なだけでほとんど同じである．なお，Δ に置き換えて $c=s$ 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 2 (a) では $\underline{s,\ c+1,\ c}$，(b) では $\underline{s+1,\ c+2,\ c}$ である．**CPA1** の整数ビット数は $\underline{\log_2(2\cdot(q_{max}-1+\rho))}$ 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 の段が上がるごとに c に 2 ずつ足した値である．(I) の c，s，k は，たとえば基数 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 $k_1, k_2, \ldots, k_{2q_{max}}$ とし，それぞれの $\widehat{rR-kD}$ の符号ビットを $m_1, m_2, \ldots, m_{2q_{max}}$ 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，$\underline{i' = \lceil \log_2(2q_{max}+2\rho+2k) \rceil}$（$k$ は $|k_1|$ と $|k_{2q_{max}}|$ の大きい方）であるが，$i'-1$ としても各 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 $m_{2q_{max}}$ 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，$-qD = \overline{m_1} \cdot (-q_{max}D) + m_1 \cdot \overline{m_2} \cdot (-(q_{max}-1)D) + \ldots + m_{2q_{max}-1} \cdot \overline{m_{2q_{max}}} \cdot (-(-q_{max}+1))D + m_{2q_{max}} \cdot (-(-q_{max})D)$ 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 $m_{2q_{max}}$ 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 $m_{2q_{max}}$ 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 $\underline{i=i'-1}$，$q_{max}=r-1$ の場合は q_{max} と $-q_{max}$ が区別できなくなるため，$\underline{i=i'}$ 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，$\widehat{rR-kD}$ 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[6] を Verilog-HDL で設計し，論理合成ツール Design-Compiler で合成して得られる速度と面積の値を用いて比較した．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．

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 $i+f+1$

Table 2 Minimum numbers of bits of **CPA2** $(i+f)$ in the case of $c=f$ of Table 1. In the case of $k=q-1/2$ the minimum is $i+f+1$.

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

設計した回路は次のとおり．54，114 ビットである．丸めは実装していない．

- 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 2 つオーバラップさせた基数 16 の構成[7]．
- r32q16_logic：基数 32，$q_{max}=16$ の論理回路実現．

図 5 に r8q4_logic の構成を示す．図 5 は図 3 にほぼ対応している．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．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．基数 8 の **MUX_l** は，論理回路実現では $q_{max}=4$ であるため 2 入力，テーブル実現では $q_{max}=7$ であるため 4 入力である．**sReg**，**Reg** はレジスタ，**Comp** は 2 の補数をとる回路，**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[8] を行う回路である．

論理回路実現に基づく提案した回路構成では，中心線 $k=q-1/2$（表 1（IV））で実装した．$k=q-1/2$ 以外（表 1（I））とすると，**CPA2** を 1 ビット小さ

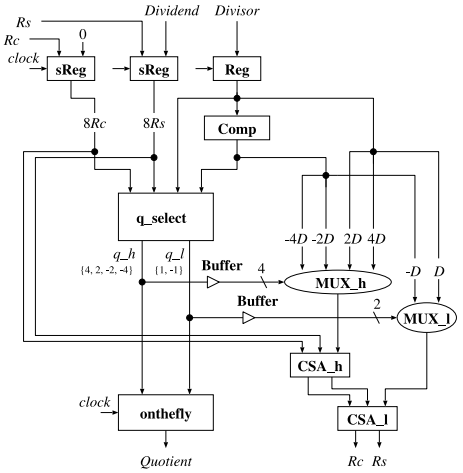

図 5 r8q4_logic の構成
Fig. 5 Hardware organization of r8q4_logic.

くできるが，$\widehat{-kD}$ を生成する回路が大きくなる．表 1（I）とした場合は，たとえば基数 4 の場合は，表 1（IV）とした場合に比べて回路量が約 5%大きくなる．

テーブル実現のテーブルはスタンダードセルで実現している．ROM や PLA で実現することもできる．テーブルの重複領域における境界線は，P-D プロットで考えると，$D = 1/2$ における重複領域の上端に近い $\widehat{rR}$ の値から始めて，各 $\widehat{D}$ の値における $\widehat{rR}$ の値はなるべく左側のそれと同じになるように選択した．同じ値を選択できない場合は，その $\widehat{D}$ の値における重複領域の上端に近い $\widehat{rR}$ の値を選択した．同じ値を選択できない場合まで待たずに，$\widehat{D}$ がもう少し切りのよい値で $\widehat{rR}$ の値を変更するようにすれば，積項の数をわずかに減らすことができる．しかし，ほとんど数字に影響しないと思われる．テーブルは case 文で記述し，論理合成ツールを用いて論理圧縮した回路になっている．

桁集合の選択は，論理回路実現では，比較の個数が少ない方が回路が小さくなるため，商の桁の最大値として冗長性が最小の $q_{max} = \lceil (r-1)/2 \rceil$ を選択した．テーブル実現では，テーブルが小さくなるように，基数 8 では商の桁の最大値として冗長性が最大の $q_{max} = 7$ を用いた．基数 4 では $q_{max} = 2$ を用いた．これは，$q_{max} = 3$ とした場合に比べてテーブルは若干大きくなるが，qD の選択の種類を D のシフトのみで作ることができ，qD を選択する MUX と，$rR - qD$ を計算する CSA を 1 段で行えるので効率が良いためである．$q_{max} = 3$ とすると $3D$，$-3D$ をフルビットで作っておき 6 入力 MUX で $3D$，$2D$，D，$-D$，$-2D$，$-3D$ を選択するか，または MUX と CSA を 2 段とする必要がある．

表 3 各実装の全体時間と面積（54 ビット）
Table 3 Total time and area of each implementation in case of 54 bit divisor.

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

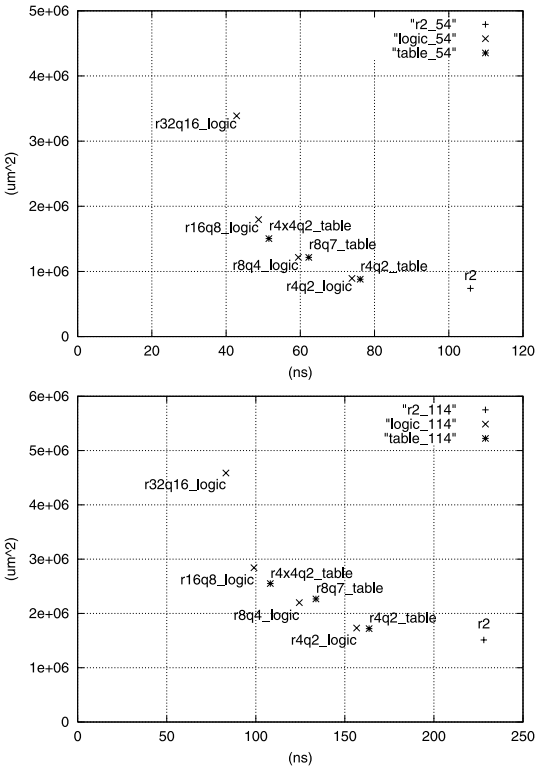

図 6 各実装の比較，54 ビット（上）と 114 ビット（下）
Fig. 6 Comparisons between implementations. In cases of 54 bit divisor (top) and 114 bit divisor (bottom).

表 3，**図 6** 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．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．同じ基数ごとに比較すると，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

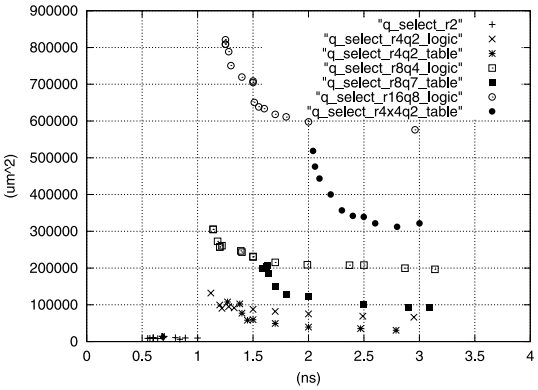

図 7 **q_select** の比較
Fig. 7 Comparison between **q_select** implementations.

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 7 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．時間は繰返しのクリティカルパス，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 6，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（I）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．**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．

除数をスケーリングする方法[6)]では論理回路実現と同様に前処理をして速度を上げている．両手法を比較する．文献 6) では，各種実装の 54 ビットでの面積と速度のデータが示されており，基数 4 のテーブル実現を基準にすると，基数 16 におけるスケーリングでは，速度が $(25 \times 28)/(29 \times 18) = 1.34$ 倍，面積が $4500/3400 = 1.32$ 倍となっている．文献 6) では，丸め（on-the-fly rounding）も実装しているため，本論文のデータとは若干実装の条件が異なる．

次に，本論文のデータを丸めを実装した場合の数値に補正する．on-the-fly の丸めを実装すると，面積では，変換回路（**onthefly**）が on-the-fly 変換のみの場合に比べて約 2 倍となる[6)]ので，54 ビットでは約 300,000 μm^2 上乗せする．速度では，丸めを行うのにサイクル数を 1 回多く数える．このようにすると，基数 16 の論理回路実現では，基数 4 のテーブル実現に比べて，速度が $(2.82 \times 28)/(3.25 \times 16) = 1.52$ 倍，面積が $2094623/1180165 = 1.77$ 倍である．

表 4 各実装のサイクル時間と面積の内訳（54 ビット）
Table 4 The details of cycle time and area of each implementation in case of 54 bit divisor.

	r4q2_logic		r4q2_table	
module	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
onthefly	-	288626	-	288626
total	2.64	892152	2.82	880165
	r8q4_logic		r8q7_table	
module	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
onthefly	-	290769	-	299342
total	3.13	1215546	3.46	1214911
	r16q8_logic		r4x4q2_table	
module	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
onthefly	-	307359	-	322997
total	3.25	1794623	3.68	1503299

6. おわりに

本論文では論理回路実現に基づく回路構成を提案し，その回路構成についてすべての基数と q_{max} における k，s，c の条件を詳細に分析し，その回路構成の有効性を示した．また，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いるため毎回の繰返しが速くなる．

謝辞 原稿を読んでいただき，貴重な助言をいただ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，貴重な指摘をしていただいた情報処理学会論文査読委員の方々に感謝いたします．本研究は東京大学大規模集積システム設計教育研究センターを通しシノプシス株式会社の協力で行われたものである．

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 (1996).
2) Robertson, J.E.: A New Class of Digital Division Methods, *IRE Trans. Electronic Computers*, Vol.EC-7, No.9, pp.218–222 (1958).
3) 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，pp.1-97–1-98 (2001).
4) Atkins, D.E.: Higher-Radix Division Using Esitimates of the Divisor and Partial Remainders, *IEEE Trans. Comput.*, Vol.C-17, No.10, pp.925–934 (1968).
5) Burgess, N. and Williams, T.: Choices of Operand Truncation in the SRT Division Algorithm, *IEEE Trans. Comput.*, Vol.44, pp.933–938 (1995).
6) Ercegovac, M.D. and Lang, T.: *Division and Square Root — Digit-Recurrence Algorithms and Implementations*, Kluwer Academic Publishers (1994).
7) Taylor, G.S.: Radix 16 SRT Dividers With Overlapped Quotient Selection Stages, *Proc. 7th IEEE Symp. Computer Arithmetic*, pp.64–71 (1985).
8) Ercegovac, M.D. and Lang, T.: On-the-Fly Conversion of Redundant into Conventional Representations, *IEEE Trans. Comput.*, Vol.C-36, No.7, pp.895–897 (1987).

付 録

A.1 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．$\Delta Min \le \Delta \le \Delta Max$，$\Delta Max-\Delta Min=\rho-1/2$ である．2 の補数の値 S，C の切捨て誤差を ΔS，ΔC，$0\le\Delta S\le 1/2^s$，$0\le\Delta C\le 1/2^c$，$c\le s$ とする．s，c は整数．

A.1.1 （I）の証明

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

$$S+C-\Delta Max\ge 0$$
$$\Rightarrow S+C-(\Delta S+\Delta C)\ge 0 \tag{10}$$
$$S+C-\Delta Min<0$$
$$\Rightarrow S+C-(\Delta S+\Delta C)<0 \tag{11}$$

が成り立てば，S，C を $s+1$，$c+1$ ビット以下切り捨てて加算した $\widehat{S}+\widehat{C}$ 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 $-\alpha\le\Delta\le 1/2^f+\beta$，$\Delta Min=-\alpha$，$\Delta Max=1/2^f+\beta$，$f=\lceil\log_2(2/(2\rho-1))\rceil$，$\alpha+\beta=(\rho-1/2)-1/2^f$，$0\le\alpha<1/2^f$，$0\le\beta<1/2^f$ 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 $s+1$，$c+1$ ビット以下を切り捨てるときに加算 $\widehat{S}+\widehat{C}$ 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，$S+C-\Delta Max\ge 0\Leftrightarrow S+C-1/2^c-\beta\ge 0\Rightarrow S+C-1/2^c\ge 0\Leftrightarrow\lfloor(S+C-1/2^c)\cdot 2^c\rfloor/2^c\ge 0\Leftrightarrow S+C-1/2^c-(((S+C-1/2^c)\cdot 2^c-\lfloor(S+C-1/2^c)\cdot 2^c\rfloor)/2^c)\ge 0\Leftrightarrow S+C-(1/2^c+((S+C)\cdot 2^c-\lfloor(S+C)\cdot 2^c\rfloor)/2^c)\ge 0\Rightarrow S+C-(\Delta S+\Delta C)\ge 0$．よって，式 (10) が成り立つ．

同様に，$S+C-\Delta Min<0\Leftrightarrow S+C+\alpha<0\Rightarrow S+C<0\Leftrightarrow\lfloor(S+C)\cdot 2^c\rfloor/2^c<0\Rightarrow$

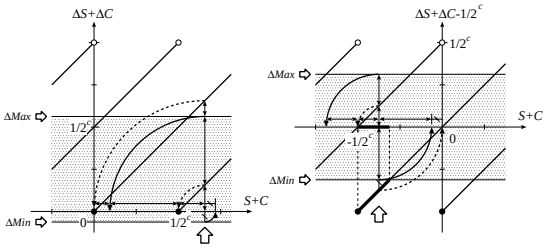

図 8 定理 1（左），2（右）の図（$c = s = f$）
Fig. 8 Diagrams illustlating Theorem 1 (left) and 2 (right) in case of $c = s = f$.

$S + C - (\Delta S + \Delta C) < 0$．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．

A.1.2 （IV）の証明

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

$$S + C - \Delta Max \geq 0$$
$$\Rightarrow S + C - (\Delta S + \Delta C - 1/2^c) \geq 0 \qquad (12)$$
$$S + C - \Delta Min < 0$$
$$\Rightarrow S + C - (\Delta S + \Delta C - 1/2^c) < 0 \qquad (13)$$

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，$\Delta S + \Delta C - 1/2^c = -1/2^\infty$）．

たとえば，$S + C = -1/2^c$，$\Delta S + \Delta C - 1/2^c = -1/2^c + ((S + C) \cdot 2^s - \lfloor (S + C) \cdot 2^s \rfloor)/2^s$ とすると，$S + C - \Delta Min = -1/2^c + 1/2^{c+1} + \alpha < 0$．$S + C - (\Delta S + \Delta C - 1/2^c) = S + C - (-1/2^c + ((S + C) \cdot 2^s - \lfloor (S + C) \cdot 2^s \rfloor)/2^s) = -1/2^c - \lfloor (-1/2^c) \cdot 2^s \rfloor / 2^s = 0$．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 8 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) は成り立たたず，$c = s = f$ とできない．

(平成 14 年 1 月 17 日受付)
(平成 14 年 6 月 4 日採録)

葛 毅

1974 年生．1997 年電気通信大学情報工学科卒業．1999 年同大学大学院博士前期課程修了．現在，同大学大学院博士後期課程在学中．コンピュータアーキテクチャ，VLSI 設計，算術演算回路のアルゴリズム等の研究に従事．

阿部 公輝（正会員）

1946 年生．1971 年横浜国立大学大学院修士課程修了．1974 年東京大学大学院博士課程単位取得退学．同年電気通信大学電子計算機学科助手．1980〜1982 年カーネギーメロン大学客員研究員．1990 年電気通信大学情報工学科助教授，現在に至る．理学博士．コンピュータアーキテクチャ，VLSI 設計，コンピュータネットワーク等の研究に従事．電子情報通信学会，IEEE 各会員．

浜田 穂積（正会員）

1938 年生．1964 年京都大学工学部数理工学科卒業．同年（株）日立製作所入社．中央研究所，システム開発研究所に所属．1989 年より電気通信大学情報工学科教授．理学博士．主な著書・編書：「PASCAL 入門」，「近似式のプログラミング」，「アルゴリズム辞典」（共編）．ACM，IEEE Computer Society，日本数学会，日本応用数理学会各会員．