計測自動制御学会東北支部 第 188 回研究集会 (2000.6.23)
資料番号 188-8

多入力直列加算回路の進化的合成

Evolutionary Synthesis of Multi-Operand Bit-Serial Adders

○寺崎俊樹*，青木孝文*，樋口龍雄*

○ Toshiki Terasaki*, Takafumi Aoki*, Tatsuo Higuchi*

*東北大学大学院情報科学研究科

*Graduate School of Information Sciences, Tohoku University

キーワード : 回路設計 (circuit design)，論理合成 (logic synthesis)，算術回路 (arithmetic circuits)，順序回路 (sequential circuits) 進化的計算手法 (evolutionary computation),

連絡先 : 〒 980-8579 仙台市青葉区荒巻字青葉 05 東北大学 大学院情報科学研究科 樋口研究室
寺崎俊樹，Tel.: (022)217-7169，Fax.: (022)263-9406，E-mail: terasaki@higuchi.ecei.tohoku.ac.jp

1. はじめに

マルチメディア時代を迎え，ディジタル信号処理向け VLSI プロセッサが多方面に応用されており，用途に応じた多種多様な演算回路の設計が要求されている．一般に，演算回路の設計は目的に応じた固有の算術アルゴリズムの知識を必要とするため，汎用の論理合成による自動設計は難しい．このため，最先端の高水準合成手法においても，高性能な算術演算コアについては，経験的に設計した機能ブロックライブラリから割り当てている．しかし，演算回路に求められる機能と性能が著しく多様化するに伴い，ライブラリベースのアプローチにも限界があると予想される．

このような設計問題を解決するため，筆者らは，演算アルゴリズムの知識を用いずに創発的に演算回路を生成する設計手法として進化的グラフ生成手法 (EGG: Evolutionary Graph Generation) を提案してきた[1)]．本手法は，個体のデータ構造として，従来の進化的計算手法に用いられる「ビット列」(GA)[2)]や「木構造」(GP)[3, 4)]ではなく，回路構造を直接表現できるデータフローグラフを用いるという特徴を有する．これまで，算術演算回路の例として定係数乗算器の合成を行い，10 ビットを越える Wallace Tree 乗算器構造の生成に成功している[1)]．ただし，これまで合成実験を行っているのは組合わせ型の算術演算回路に限定されていた．そこで，本稿では，より一般的に順序回路型の算術演算回路を対象として，EGG による創発生成を試みる．

EGG システムにより実用的な規模の順序回路を合成する場合，進化的計算手法で一般に問題となっている計算時間に関する問題を解決する必要がある．EGG システムにおける実行時間は，主に，世代毎に生成された回路グラフの機能評価に費やされている．従来のテストパターンの入出力関係による手法では，順序回路の機能を検証する場合，入力数を n，入力語長を t として $O(2^{nt})$ 時

間が必要となる．本稿では，演算回路の機能検証にかかる計算時間を削減する手法として，記号検証手法を用いた回路評価方法を提案する．この記号検証手法は，与えられた回路構造に対応する数学的な方程式を導出することにより，与えられた演算回路の機能を高速に求める手法である．本稿では，記号検証手法を順序回路に応用した例として，多入力直列加算回路の合成を取り上げる．順序回路合成に拡張された EGG システムでは 8 入力直列加算回路をおよそ 1 時間で合成可能である．また，提案する手法は直列加算回路に限らず，様々な順序回路の合成に応用できる．その一つとして，積和演算回路の合成例を本稿の最後に示す．

2. EGG システムの基本概念

進化的グラフ生成手法 (EGG) は，進化論的計算手法[5]に基づき，グラフ構造の最適化を効率よく行う手法である．一般に，進化論的計算手法とは，与えられた環境に適応する複雑な構造が世代を重ねることで出現するという自然進化の過程を模擬したものである．この手法における進化では，与えられた条件下で適応した個体ほど集団中に長く存在し，進化的操作によって多くの子孫を残す機会が与えられる．そのため，世代が進むにつれ，より良い構造が個体群中に広がり，結果として，個体群中には設計者が定めた制約を満たす個体が生成される．EGG システムでは個体表現として，回路構造をモデル化したデータフローグラフそのものを扱い，グラフ論的に定義された進化的操作により個体群を進化させる．従って，GA のビット列によるモデル化や，GP の木構造によるモデル化のような非直接的な置き換えを行う必要がなく，EGG はグラフ構造をより直接的に操作するように設計されているといえる．

EGG では回路構造を Fig.1に示す回路グラフで

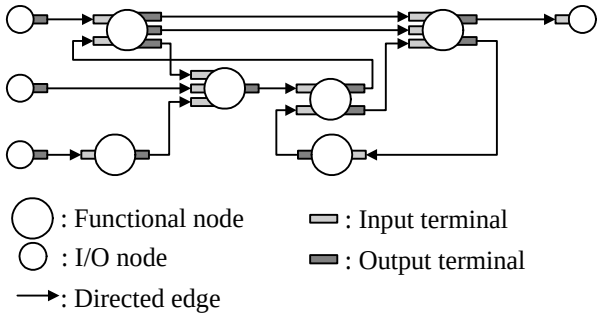


Fig. 1　Example of a circuit graph.

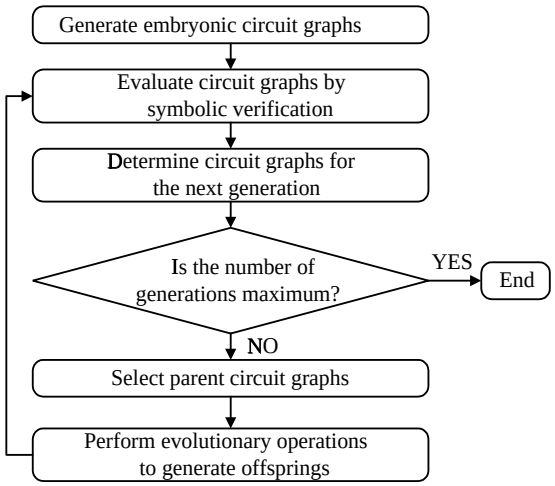


Fig. 2　EGG system flow.

表現する．回路グラフ G は次のように定義される．

$$G = (N(G), D(G)) \tag{1}$$

ここで，$N(G)$ はノードの非空有限集合であり，$D(G)$ は有向辺の集合である．ノードには機能ノード及び入出力ノードがあり，各ノードには機能，複数の入出力端子の属性を持つ．有向辺は必ずノードの出力端子から入力端子に向かうものとし，出力端子と入力端子は常に一対一で対応するものとする．ここで，未接続の端子が存在しない回路グラフを，特に完全回路グラフと呼び，EGG では完全回路グラフのみを扱うものとする．

Fig.2に EGG システム全体のフローを示す．進

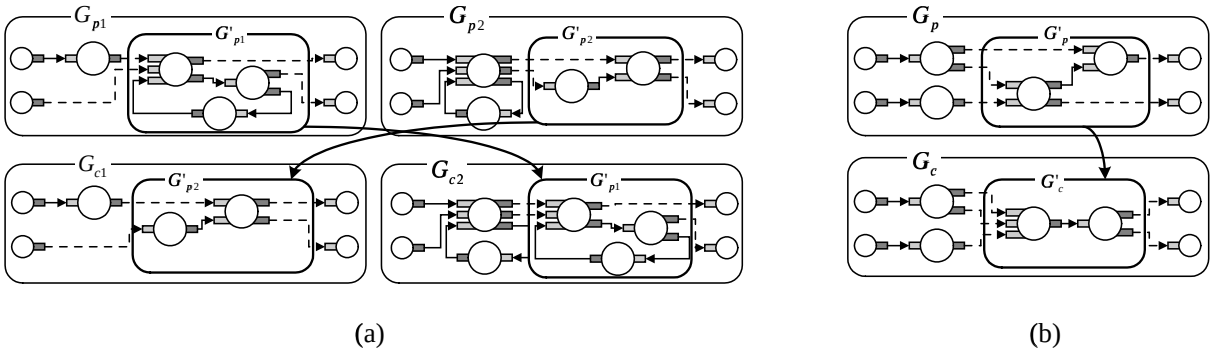


Fig. 3　Examples of evolutionary operations: (a) crossover, and (b) mutation.

化的操作が行われた個体群内の個々の回路グラフは，記号検証手法（3.章参照）を用いて評価される．高い評価を与えられた回路グラフに対しては進化的操作が行われ，次の世代の個体群となる子孫が生成される．EGG には進化的操作として「交叉」と「突然変異」があり，この 2 つにより回路グラフの構造操作を行う.「交叉」は，2 つの回路グラフの部分回路グラフを交換して新たな回路グラフを作り出す操作である（Fig.3(a)）．一方,「突然変異」は個体構造の一部をランダムに生成した構造に置き換える操作である（Fig.3(b)）．2 つの操作は，完全性を保存しながら回路グラフの構造を変更する．選択された個体が完全回路グラフであれば，進化的操作後に生成される回路も完全回路グラフとなる．

3.　多入力直列加算回路の設計

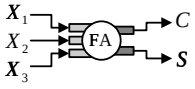
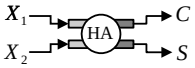
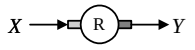
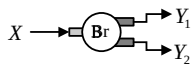
本章では，順序回路の合成に拡張した EGG システムについて記述する．システムの評価関数を変更することにより，容易に他の設計仕様に適用できるという EGG の特徴に注目している．

EGG システムにより生成された回路グラフは「機能評価」と「性能評価」という 2 つの評価基準により評価される．機能評価 F は求める機能と比較して，どの程度求める機能を満たしているかを評価するものであり，性能評価 P は回路遅延 D とモジュール間配線数 A との積から与えられる関数である．

まず，機能評価 F について説明する．機能評価は，EGG システムにより生成された回路グラフの機能を検証し，結果を評価することにより与える．これまで，回路の機能は，すべての回路グラフを Verilog-HDL コードに変換し，シミュレーションを行うことにより検証されていた[1)]．このテストパターンによる順序回路の機能検証には，入力数 n，入力語長 t として $O(2^{nt})$ 時間を要する．EGG システム全体の計算時間は，この時間に個体数及び世代数をさらに掛け合わすことにより求められる．EGG を実際の設計問題へ応用する場合，この計算時間は大きな問題となることは明らかである．拡張した EGG システムでは記号検証手法を用いることにより，この問題を解決している．本稿では，数学的な方程式を連立し，解くことにより算術回路の機能を高速に検証する手法を提案する．この手法により機能検証時間を $O(m^2)$ 時間に削減可能である．ここで m は回路グラフ内のノード数である．

以降，1 ビット目が 2^0，2 ビット目が 2^1，3 ビット目が 2^2 のような重み付き符号なし 2 進数系を用いた LSB-first の Bit-serial 演算を仮定する．すべての入力信号は非負整数である．今，Fig.4に示される直列演算回路の記号検証を考える．Table 1に示される機能ノードの方程式表現を用いることによ

Table 1　Functional nodes used in the experiment.

Name	Symbol	Mathematical representation	Delay
Full adder		$2C + S = X_1 + X_2 + X_3$	2τ
Half adder		$2C + S = X_1 + X_2$	τ
1-bit register		$Y = 2X$	——
Branch		$Y_1 = X, Y_2 = X$	0

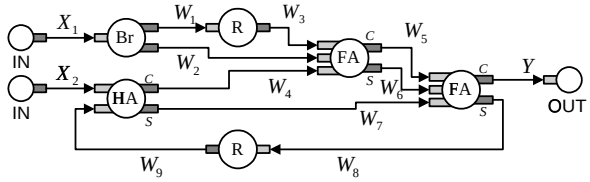


Fig. 4　Example of a 2-input bit-serial arithmetic circuit.

り，Fig.4の回路構造は次に示す連立方程式で表現できる．

$$\begin{aligned} W_1 &= X_1, \\ W_2 &= X_1, \\ W_3 &= 2W_1, \\ 2W_5 + W_6 &= X_2 + W_3 + W_4, \\ 2W_4 + W_7 &= X_2 + W_9, \\ W_9 &= 2W_8, \\ 2Y + W_8 &= W_5 + W_6 + W_7 \end{aligned}$$

ここで，変数$X_1, X_2, Y, W_1, \cdots, W_9$は非負整数の変数で，Fig.4に示される有向辺上のデータに一致する．回路の入出力関係は与えられた方程式をガウスの消去法などを用いて解くことにより得られる．

$$2Y = 3X_1 + X_2 - W_4 - W_5 + W_8$$

この例で示されるように，Table 1のノードで構成されるn入力1出力直列算術回路の機能は，一般に以下の方程式で表現できる．

$$\hat{K}_0 Y = \sum_{i=1}^{n} \hat{K}_i X_i + f(X_1, \cdots, X_n) \qquad (2)$$

ここで，$X_i\ (i = 1, \cdots, n)$はビット列の入力，Yは出力，$\hat{K}_i\ (i = 0, \cdots, n)$は非負整数定数，$f(X_1, \cdots, X_n)$は入力に対する非線型関数を表す．特に，$f$はガウスの消去法によって消去できなかった中間変数からなる項である．

目標とする回路の方程式表現を，非負整数係数$K_i\ (i = 0, \cdots, n)$を用いて，

$$K_0 Y = \sum_{i=1}^{n} K_i X_i \qquad (3)$$

とする．特に，n入力直列加算回路を合成する場合，目標とする係数は$K_0 = K_1 = \cdots = K_n = 1$となる．機能評価$F$は式(2)における係数$\hat{K}_i$と目標とする係数$K_i (i = 0, 1, \cdots, n)$との類似度を評価するものである．この類似度を求めるために，まず，係数値を2進数で表現する．

$$\begin{aligned} \hat{K}_i &= \hat{k}_{i,0}2^0 + \hat{k}_{i,1}2^1 + \hat{k}_{i,2}2^2 + \cdots \\ &\qquad + \hat{k}_{i,\|\hat{K}_i\|-1}2^{\|\hat{K}_i\|-1}, \\ K_i &= k_{i,0}2^0 + k_{i,1}2^1 + k_{i,2}2^2 + \cdots \\ &\qquad + k_{i,\|K_i\|-1}2^{\|K_i\|-1} \end{aligned}$$

ここで，$\|x\| = \lceil \log_2(x+1) \rceil$である．これらの係数の類似度は，相関値から求められる．シフト

量を s とすると，2 つの係数のビット列の相関値 $M_{\hat{K}_i,K_i}(s)$ は次式で与えられる．

$$M_{\hat{K}_i,K_i}(s)=\begin{cases}\dfrac{1}{\|\hat{K}_i\|}\displaystyle\sum_{l=0}^{\|\hat{K}_i\|-1}\delta\left(\hat{k}_{i,l}-k_{i,l-s}\right) \\ \qquad \text{if} \quad \|\hat{K}_i\|\geqq\|K_i\|\,, \\ \dfrac{1}{\|K_i\|}\displaystyle\sum_{l=0}^{\|K_i\|-1}\delta\left(\hat{k}_{i,l-s}-k_{i,l}\right) \\ \qquad \text{if} \quad \|\hat{K}_i\|<\|K_i\|\end{cases} \tag{4}$$

ここで，$\delta(x)$ は，

$$\delta(x)=\begin{cases}1 & \text{if} \quad x=0, \\ 0 & \text{if} \quad x\neq 0\end{cases} \tag{5}$$

である．上記の計算において，定義されていない桁には常に 0 を与えるものとする．この相関関数を用いて，式 (2) と式 (3) の類似度 F'を次式で定める．

$$F'=\frac{1}{n+1}\sum_{i=0}^{n}\left[\max_{0\leqq s\leqq d}\left\{100M_{\hat{K}_i,K_i}(s)-C_1 s\right\}\right], \tag{6}$$

ここで，$d=\left|\|\hat{K}_i\|-\|K_i\|\right|$ であり，本実験では $C_1=10$ としている．項 $C_1 s$ はシフト量 s に比例した負の影響を表している．この類似度 F'を用いて，機能評価 F は次式で定める．

$$F \quad = \quad F'-C_2 q, \tag{7}$$

ここで，q は回路を展開したときの Delay-free loop 数を表しており，本実験では $C_2=5$ としている．

性能評価 P は次式で与えられる．

$$P=\frac{C_3}{DA}, \tag{8}$$

ここで，A はモジュール間配線数，D は 2 入力 XOR ゲートを単位遅延としたときのレジスタ間の最大遅延である．回路グラフの適合度は $F+P$ で与えるが，C_3は P_{max}/F_{max}が 5/100 程度となるように設定されている．

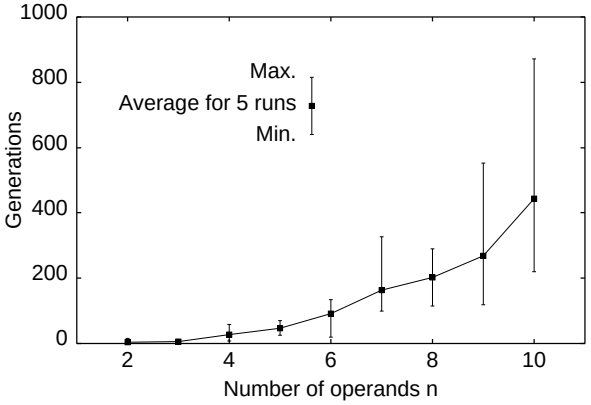
4. 実験結果

本実験では式 (3) において $K_0=K_1=\cdots=K_n=1$ で与えられる n 入力直列加算回路を目標として合成実験を行った．本実験で使用した主要なパラメータを Table 2 に示す．Fig.5は EGG システムによる n 入力直列加算回路 $(2\leq n\leq 10)$ の合成結果である．個々の入力数 n に対して 5 回の独立した進化的合成を行った．Fig.5(a) は 100%機能を満たす回路が得られたときの平均世代数を示しており，Fig.5(b) は 3000 世代後に得られた最大の DA 積の 5 回の平均値を示している．また，エラーバーは 5 回の試行におけるばらつきの幅を示している．図より，EGG システムは最適な 8 入力直列加算回路を 3000 世代以内に合成できることがわかる．これは Sun Ultra 60 のワークステーション (CPU:360MHz, Memory 1.15GB) で約 1 時間に相当する．2 入力から 10 入力の直列加算回路に対してそれぞれ 5 回の試行を行ったときに得られた最良個体を Fig.6に示す．

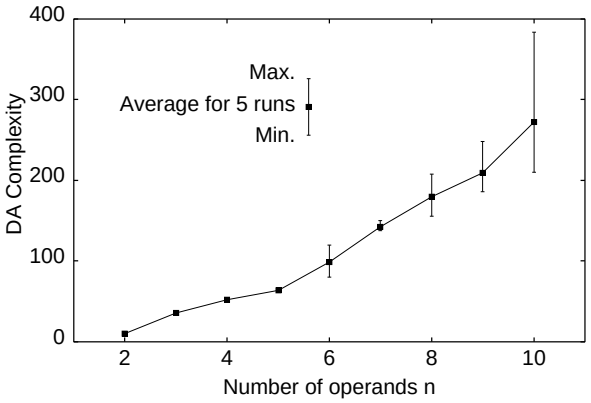
Table 2　Main parameter values.

Population size	100
Max. num. of generations	3000
Max. num. of nodes	30
Crossover rate	0.7
Mutation rate	0.1

以降，8 入力直列加算回路の合成を一例として，EGG における進化の過程をさらに詳細に記述する．Fig.7は 20 回の試行における最良個体の適合度の推移を表したものである．個々の試行において最良個体の適合度が階段状に上昇していることがわかる．1 つの個体群の進化の様子を詳しく示したものが Fig.8である．縦軸は世代数，横軸は機能評価 F 及び DA 積を表している．ランダムな初期世代が与えられると，まず，進化は機能評価値を上げる方向へ進む．個々の個体は目標とする機能に一致する特定の DA 積を保持しながら進化する傾向にあることがわかる．122 世代目で機能を 100%満たす個体が得られている．この個体の DA



(a)



(b)

Fig. 5 Result of five evolutionary runs: (a) the number of generations required to obtain the first individual having 100% functionality, (b) the best DA product obtained in the 3000th generation.

積は 290 であった．3000 世代後に Fig.6(g) に示す最良の加算回路の構成が得られた．DA 積の値が 156 に減少しており，この構造は最小段数で構成されていることがわかる．以上から，算術アルゴリズムに関する特別な知識を用いずに進化的に順序算術回路を合成できることが示された．

提案する手法は，評価関数を変更することにより，多入力直列加算回路の合成だけでなく，ほかの設計問題にも適用することができる．例えば，目標とする方程式 (3) の変数を $n = 2, K_0 = 1, K_1 = 3, K_2 = 5$ とした場合，$Y = 3X + 5Y$ という方程式で与えられる積和演算器を合成できる．Fig.9は

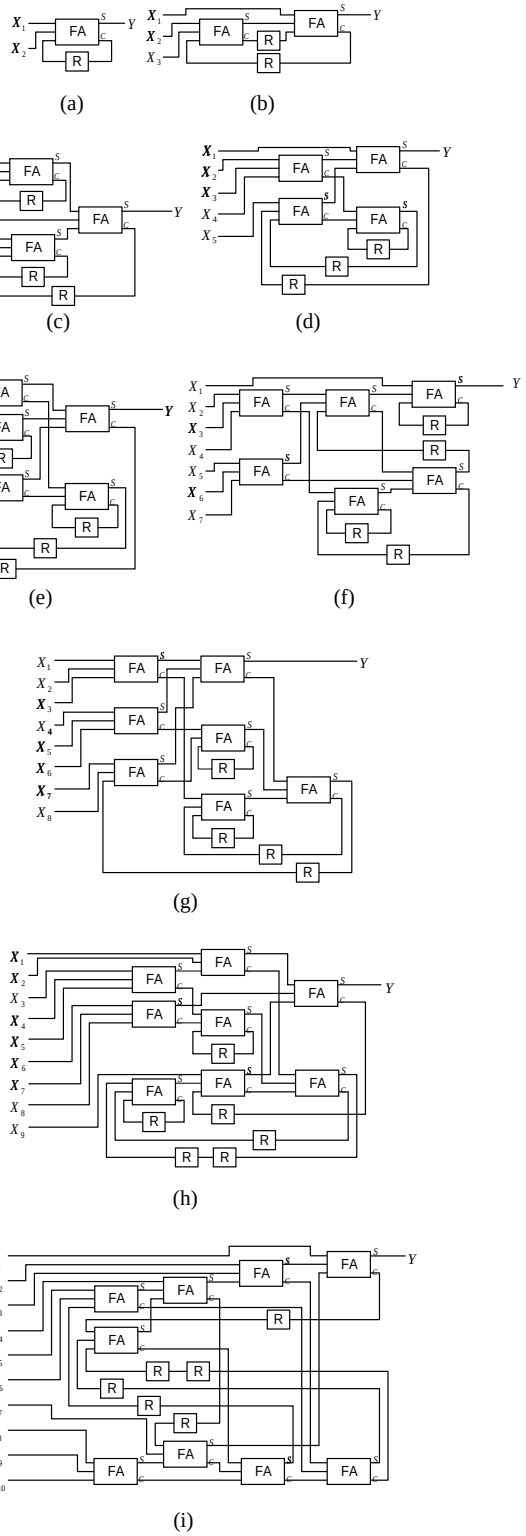


Fig. 6 Best individuals obtained in the 3000th generation, where the number of operands are (a) n=2, (b) n=3, (c) n=4, (d) n=5, (e) n=6, (f) n=7, (g) n=8, (h) n=9, and (i) n=10.

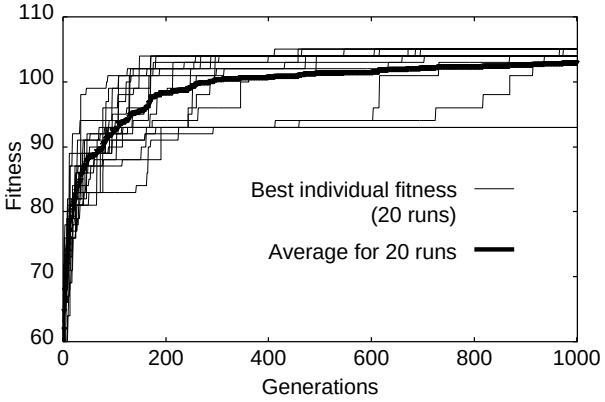


Fig. 7 Transition of the best individual fitness in the population.

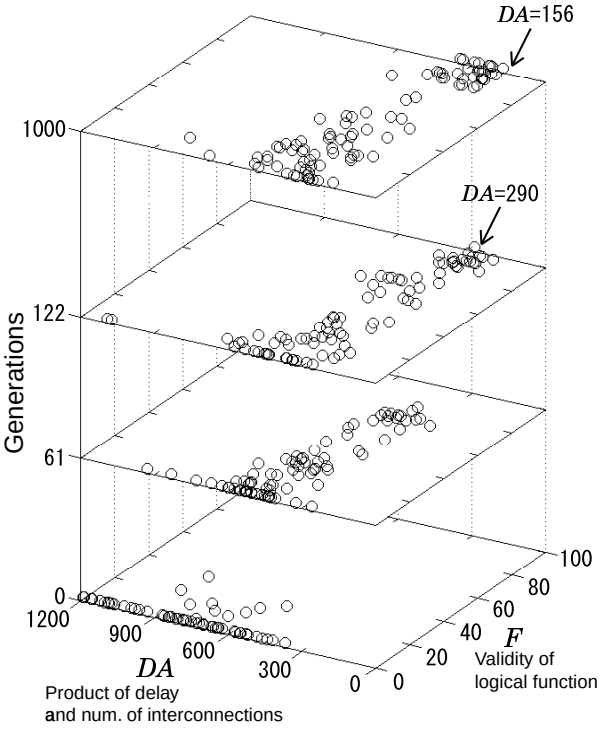


Fig. 8 Example of the evolution process of an 8-operand adder.

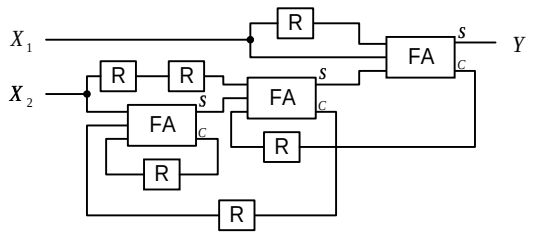


Fig. 9 Evolved multiply-addition structure under the target function : $Y = 3X_1 + 5X_2$.

3000 世代で得られた最良の回路である．更なる研究が必要ではあるが，一般的な設計問題を扱えるEGG ベースの算術回路合成システムを構築することは可能であると考えられる．

5. おわりに

本稿では，EGG システムの多入力直列加算回路への応用を示した．進化の過程で生成された算術回路の機能を高速に評価する手法として，記号的検証手法を用いた新しい回路の機能検証手法を提案した．多入力直列加算回路の実験的な合成から，算術アルゴリズムに関する特殊な知識を用いずに順序算術回路を生成できるという EGG システムの潜在的な能力が示された．現在，並列 EGG システムのための専用 PC クラスタシステム構築に向けた研究を行っている．

謝辞

本稿を取りまとめるにあたり，親切なご助言を頂いた東北大学博士後期課程本間尚文氏に深く感謝致します．

参考文献

1) Aoki, T. and Homma, N. and Higuchi, T.: Evolutionary Design of Arithmetic Circuits, IEICE Trans. Fundamentals, **E82-A**-5, 798/806 (1999)

2) D. Quagliarella, J. Periaux, C. Poloni, G. Winter: Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, 105/131, JOHN WILEY & SONS (1997)

3) Koza, R. J. and III, Bennett, H. F. and Andre, D. and Keane, A. M. and Dunlap, F.: Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming, IEEE Trans. Evolutionary Computation, **1**-2, 109/128 (1997)

4) 北野 宏明 編著: 遺伝的アルゴリズム 2，207/249，産業図書 (1995)

5) Back, T. and Hammel, U. and Schwefel, P. H.: Evolutionary Computation: Comments on the History and Current State, IEEE Trans. Evolutionary Computation, **1**-1, 3/13 (1997)