

LC_001

入力粒度に依存しない再構成論理セルのフィジビリティスタディ Feasibility Study of Variable Grain Logic Cell for Reconfigurable Device

尼崎 太樹† 山口 良一† 中山 英明† 飯田 全広† 末吉 敏則†
Motoki Amagasaki Ryoichi Yamaguchi Hideaki Nakayama Masahiro Iida Toshinori Sueyoshi

1. はじめに

従来のRLD (Reconfigurable Logic Device) に見られるアーキテクチャは、論理ブロックで扱う入力粒度の違いにより、細粒度方式と粗粒度方式の2種類に大別できる[1]。細粒度方式では、従来のFPGA (Field Programable Gate Array) に用いられるLUT (Look-Up Table) やマルチプレクサ等、入力粒度の小さいセルで構成されており、ビット単位の入力に対して任意の論理演算を実装可能である。一方、粗粒度方式ではALU (Arithmetic and Logic Unit) 等、入力粒度の大きい演算回路を最小構成にしているため、バイト単位の算術演算を高速に実行するのに適している。

しかしながら、実装するアプリケーションには、処理の内容に応じて粗粒度方式に適したものと細粒度方式に適したものが存在する。例えば、アドレスデコードの場合ほとんどが細粒度処理であり、デジタルフィルタの場合は大半を粗粒度処理が占める。このため、ALUやMAC (Multiply and ACcumulation) 演算等の算術演算回路を細粒度方式で実装した場合、専用ハードウェアの数十倍程度の面積を要し、動作速度も一桁遅い。一方、粗粒度方式でも、演算器間のグルー・ロジックでオーバーヘッドが生じるといった特徴がある。したがって、従来のRLDは、不得意な演算を補うための専用回路を論理ブロックの外に搭載することで性能低下を回避している。しかし、これらの専用回路は、使用されない場合には、逆にチップ上の実装効率を落とす一因となる。

以上より、既存のデバイスでは実装面積と動作速度の両立が難しく、使用するデバイスが実装するアプリケーションの種類を限定しているのが現状である。これよりRLDの性能改善のためには、1つ、もしくは複数の論理セルで、算術演算と論理演算共に高い実装効率を実現することが重要になる。そこで我々は、回路実装効率の問題を改善するために、細粒度方式と粗粒度方式の特徴を組み合わせた粒度可変構造論理セルアーキテクチャを提案している[2][3]。提案論理セルでは、論理ブロック毎に入力粒度の切り替えを行うことで、アプリケーションに左右されることなく、高い面積効率と動作速度が期待できる。本稿では実際に、ベンチマーク回路を配置配線し、面積及び速度評価を行う。

2. 提案論理セルアーキテクチャ

入力粒度に依存せず算術演算と論理演算に適した論理セルを考えるにあたり、我々は全加算器と2入力Reed-Muller標準形に共通点に着目する。図1(a)は1ビット全加算器の構成を示し、(b)は2入力Reed-Muller標準形を示す。2入力Reed-Muller標準形は4ビットのコンフィギュレーションメモリを備え、従来の2-LUTと同様の出力論理を実現できる。2入力Reed-Muller標準形

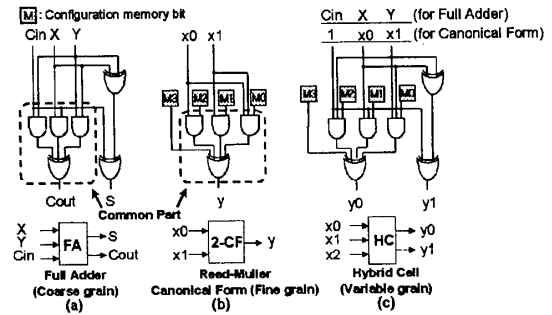


図1: Structure of hybrid cell.

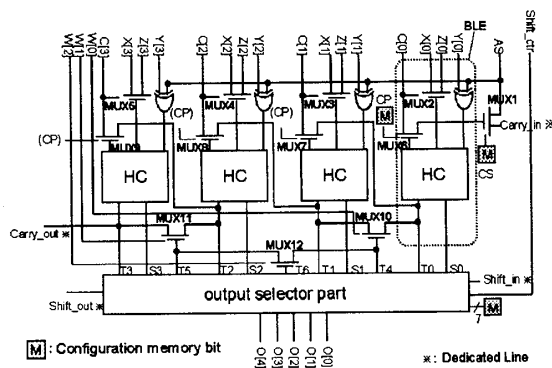


図2: Variable Grain Logic Cell Architecture.

は式(1)で表現され、中括弧で囲まれる部分がそれぞれのコンフィギュレーションメモリビットに対応する。

$$F(x_0, x_1) = \{ F(0, 0) \oplus x_0 \{ F(0, 0) \oplus F(1, 0) \} \oplus x_1 \{ F(0, 0) \oplus F(0, 1) \} \oplus x_0 x_1 \{ F(0, 0) \oplus F(0, 1) \oplus F(1, 0) \oplus F(1, 1) \} \} \quad (1)$$

(a) (b) 2つの回路を比較すると、それぞれが出力段のEXORとそれにつながるAND回路を持つことがわかる。したがって(a)の全加算器に4ビットのメモリを付加することで、(c)のように全加算器と2入力Reed-Muller標準形が表現可能なハイブリッドセル (Hybrid Cell, 以下HC) を構成する。

ここで我々は、HCを複数組み合わせることで、より多彩な機能を持つ粒度可変構造論理セル (Variable Grain Logic Cell, 以下VGLC) を提案している[2]。図2にVGLCの構成を示す。VGLCは、HCとその入力を制御する前置回路でBLE (Basic Logic Element) を構成し、4つのBLEと出力選択部、25ビットのコンフィギュレーションメモリよりなる。

続いて図3に、VGLCが持つ5種類の基本機能を示す。まず最初に、BLEを全加算器として使用すると、VGLC

†熊本大学大学院 自然科学研究科

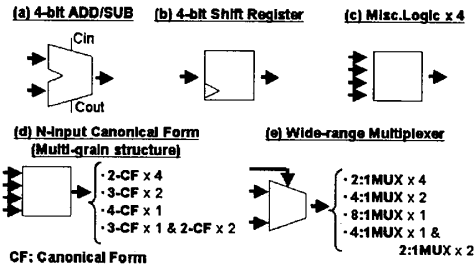


図 3: Basic function of the VGLC.

表 1: Logic pattern of BLE (CP = 0).

AS	4 input variable		3 input variable	
	T 端子	S 端子	T 端子	S 端子
0	182/65,536	24/65,536	120/256	43/256
1	230/65,536	24/65,536	148/256	43/256
Total	446/65,536 (0.68%)		206/256 (80.47%)	

は4ビットの加減算器を構成する。キャリイパスは外部配線を通ることなくVGLC内でクラスタ結合されているため、キャリイの伝播は高速に行われる。また、VGLC間のキャリイパスは専用ラインで接続されており、演算に応じてビット幅を拡張可能である。2番目に、VGLCの出力選択部は、4個のフリップフロップ(Flip Flop, 以下FF)とセレクタで構成される。この回路は4ビットシフトレジスタとしても動作し、加算器と同様にビット幅を拡張可能である。3番目に、1つのBLEは各BLEに共通な信号ASを除くと、4入力2出力のボックスと見ることができる。これより、自身のゲート構造を利用して、4入力までの限られた論理を表現できる。表1に、1個のBLEで表現可能な出力論理パターンを示す。4変数と3変数入力での全出力論理数はそれぞれ65,536パターン、256パターン存在する。例えば、AS=0とした場合、BLEのT端子より120パターンの出力論理が表現可能であることを表している。これよりS端子とT端子を合わせた場合、2入力の論理標準形相当の面積とコンフィグレーションメモリ数で、3入力論理の81.3%、4入力論理の0.68%が表現可能である。我々はこの論理の表現方法をMisc.Logic機能と名付ける。図4に、BLEを3入力論理(x_2, x_1, x_0)のMisc.Logicとして用いた場合の例を示す。4番目に、BLEを2入力標準形として用いた場合、図2のMUX10-12を使って3入力と4入力論理の標準形を実装できる。これよりVGLCは、2-4入力までのマルチグレイン構造を取ることができる。最後に、BLEは2to1MUXとしての機能を持つので、上記と同様にMUX10-12を用いて、最大8入力のワイドレンジマルチプレクサを構成する。

以上よりVGLCは、算術演算時には、4ビットリップルキャリアダーやワイドレンジマルチプレクサとして動作する。これより、従来の粗粒度方式の論理ブロックと同等の演算性能が期待できる。一方、論理演算時には、Misc.Logicや2-4入力論理の標準形として、細粒度方式と同様の柔軟性を持つことができる。以上より提案論理セルは、粗粒度タイプ向きの回路も、細粒度タイプ向きの回路も演算にあわせた実装が可能となり、デバイスの

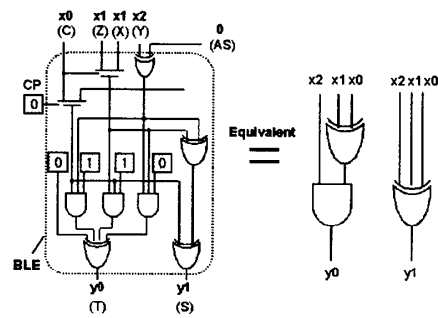


図 4: Example of usage of the Misc.Logic function.

表 2: Mapping factor of each architecture.

Function	$C_{trans.}$	C_{bit}
	VGLC	
2入力標準形	154	7
3入力標準形	308	13
4入力標準形	616	25
Misc.Logic	154	7
LUT + FF		
3-LUT	110	9
4-LUT	206	17
5-LUT	398	33

性能を最大限に活用できると期待が持たれる。

3. Misc.Logicの有効性評価

実際にベンチマーク回路を3入力論理でテクノロジマッピングし、Misc.Logic機能でカバー可能な論理を抽出する。カバー不可な論理に関しては、2, 3入力論理の標準形を用いて実装する。表2にそれぞれのファンクションを実装するのに必要なトランジスタ数 $C_{trans.}$ とコンフィグレーションメモリ数 C_{bits} を示す。この係数をマッピングに要するセル数に掛けることで、総トランジスタ数と総メモリ数を算出する。LUTに要する係数は、トランスミッションゲートでLUTを構成した場合[4]を想定している。ベンチマーク回路にはOPENCORE[5]より10種類の回路を用いる。

表3にマッピング結果を示す。提案論理ブロックにおいて、(A) Misc.Logicを用いた場合、(B) 2, 3入力の標準形のみで実装した場合と比較して、トランジスタ数、コンフィグレーションメモリ数共に6, 7割程改善されることがわかる。(C) (D) (E)と比較しても、総Tt数では3-LUTに劣るものの、4, 5-LUTにおいては同等以上となる。また、コンフィグレーションメモリ数では大幅に改善される。このように、論理演算実装時には、Misc.Logic機能を用いたマッピングが有効である。

4. 実装面積と動作速度評価

4.1 評価フローと面積モデル

実際にベンチマーク回路を配置配線し、4-LUTベースのロジッククラスタ(以下LC)と実装面積、動作速度の2点で比較する。この時、0.35- μm , 0.18- μm , 0.07- μm 3種類のプロセスに関して評価を行う。配置配線ツールには、トロント大で開発されたVPR[6]を用いる。ただし、我々のアーキテクチャは既存の論理合成ツールやテクノロジマップには対応していないため、VPRの入力ネットリストをハンドマッピングで作成する。このため、規模

表 3: Mapping result of each architecture.

Benchmark circuit	(A).VGLC(MISC)		(B).VGLC(no Misc.)		(C).3-LUT		(D).4-LUT		(E).5-LUT	
	#Tr.	#Bits	#Tr.	#Bits	#Tr.	#Bits	#Tr.	#Bits	#Tr.	#Bits
adder128	184k	8k	295k	13k	131k	11k	186k	15k	306k	25k
adder256	375k	17k	602k	26k	268k	22k	378k	31k	612k	51k
aes_core	1,338k	60k	2,411k	102k	900k	74k	1,214k	100k	1,138k	94k
biquad	327k	15k	561k	24k	233k	19k	346k	29k	578k	48k
cf_fir	6,485k	295k	10,770k	462k	4,629k	379k	7,786k	642k	12,444k	1,032k
gpio	231k	10k	394k	17k	161k	13k	153k	13k	332k	28k
jpeg_encoder	6,747k	306k	11,625k	496k	4,773k	391k	7,380k	609k	12,010k	996k
spl	297k	13k	484k	21k	197k	16k	275k	23k	412k	34k
tv80	645k	29k	1,133k	48k	447k	37k	614k	51k	972k	81k
vga_lcd	9,582k	435k	16,969k	723k	6,836k	559k	7,532k	622k	11,842k	982k
Ave.	2,621k	119k	4,524k	193k	1,858k	152k	2,586k	213k	4,065k	337k
ratio of (A)	-	-	173 %	162 %	71 %	128 %	99 %	179 %	155 %	283 %

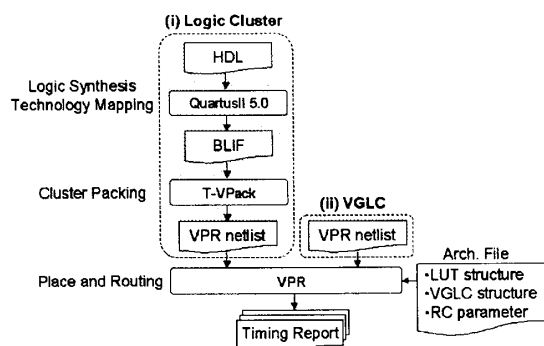


図 5: Architecture evaluation flow.

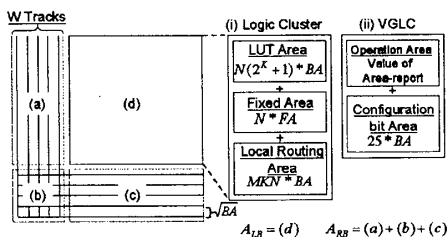


図 6: Area model.

は小さいが、32ビット乗算器と8ポイントFFT(8ビット精度)をベンチマーク回路として用いる。32bit乗算器は2次のBooth法で構成する。図5に評価フローを示す。

本評価には、図6に示す面積評価モデルを用いる。本モデルは、規則的に配置され独立した論理ブロック部(A_{LB})と、水平・垂直方向の配線部(A_{RB})よりなる。各トラックの幅はビット面積 BA の平方根とし、論理ブロックは正方形と仮定する。表4に各パラメータの定義を示す。

$$A_{Tile} = A_{LB} + A_{RB} \quad (2)$$

$$A_{LB} = N\{(2^K + 1)BA + FA\} + MKN \times BA \quad (3)$$

$$A_{RB} = W^2BA + 2W\sqrt{A_{LB}}\sqrt{BA} \quad (4)$$

このモデルは、トロント大Roseらのモデル[7]に、クラスタ数 N とLCのローカルルーティングに要するメモリ数 M を付け加えたものである。 W はVPRより得ら

表 4: Parameters of the area model.

パラメータ	説明
A_{tile}	論理ブロックと配線領域を含めた面積
A_{LB}	論理ブロックの面積
A_{RB}	配線領域の面積
N	ロジッククラスタ数
K	LUTの入力粒度
BA	メモリの単位面積
W	配線トラック数
M	ローカル配線の選択に必要なメモリ数
FA	FF及びその他固定回路が占める面積

表 5: Area and speed estimation parameter.

	VGLC	Logic Cluster (LC)		
		$N = 1$	$N = 4$	$N = 8$
$A_{LB}[k\lambda^2]$	348.41	42.75	203.78	440.32
$T_{BLE}[ns]$	2.8	1.6	1.6	1.6

れた値を用い、 $N=1, 4, 8$ とする。ただし、提案アーキテクチャにおいては、Rohm社0.35- μm スタンダードセルライブラリを用いて合成を行い、得られた面積とコンフィギュレーションメモリの面積を合計し A_{LB} とする。

加えて、プロセスをまたいだ面積評価を行うため、 λ^2 (λ はプロセス特徴量の半分値[8])を用いて、面積をグリッド換算する。これにより、すべてのアーキテクチャは同じプロセスで製造された場合の関係を求めることができる。この時、 BA と FA は文献[7]より、それぞれ13,056[λ^2]、1,024[λ^2]となる。以上をもとに、表5を面積、遅延評価に関するパラメータとして用いる。LCの組み合わせ遅延 T_{BLE} は、VGLC同様、0.35- μm プロセスで製造されたXC4000[9]より参照する。0.18- μm 、0.07- μm プロセスに関しては、この値をスケールしたものをを用いる。VPRで用いる配線時のRCパラメータは、BPTM[10]より得た値を各プロセスごとに使用する。

4.2 評価結果と考察

32ビット乗算器と8ポイントFFTを実装した結果をそれぞれ表6、表7に示す。ここでの N_{LB} はマッピングに必要な論理ブロック数を指す。この結果、提案アーキテクチャのクリティカルパス遅延は他のLCよりも大きい。総面積は比較アーキテクチャの中で最も小さくなる。遅延が大きいのは、VGLCをスタンダードセルで

FIT2006 (第5回情報科学技術フォーラム)

表 6: Total Area and Delay in 32-bit multiplier.

	0.35- μm				0.18- μm				0.07- μm			
	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]
VGLC	23	17 \times 17	508.34	432.31	20	17 \times 17	437.42	253.83	22	17 \times 17	484.11	123.34
LC(N=1)	15	47 \times 47	1041.87	312.39	14	47 \times 47	947.04	150.89	14	47 \times 47	947.04	83.29
LC(N=4)	33	24 \times 24	1308.85	230.34	34	24 \times 24	1365.00	129.85	34	24 \times 24	1365.00	64.60
LC(N=8)	46	17 \times 17	1318.03	219.50	44	17 \times 17	1240.21	117.80	45	17 \times 17	1278.82	70.66

表 7: Total Area and Delay in 8-point FFT.

	0.35- μm				0.18- μm				0.07- μm			
	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]	W [-]	N_{LB} [-]	Total Area [$M\lambda^2$]	Delay [ns]
VGLC	21	19 \times 19	575.19	222.62	20	19 \times 19	546.39	122.65	20	19 \times 19	546.39	58.39
LC(N=1)	13	54 \times 54	1130.93	208.05	13	54 \times 54	1,130.93	107.44	12	54 \times 54	1017.70	55.77
LC(N=4)	31	27 \times 27	1518.83	166.62	29	27 \times 27	1,387.13	100.63	32	27 \times 27	1586.92	59.10
LC(N=8)	41	19 \times 19	1408.93	172.21	43	19 \times 19	1,501.70	105.58	40	19 \times 19	1363.66	68.44

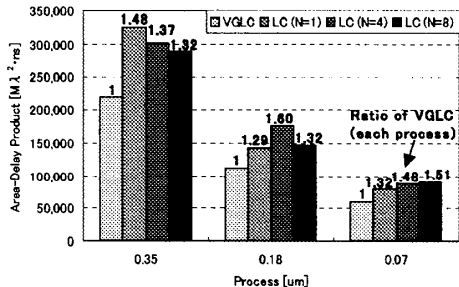


図 7: Area-Delay Product (32-bit multiplier).

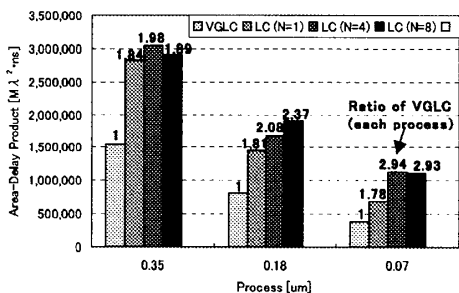


図 8: Area-Delay Product (8-point FFT).

実装していることが一因と考えられる。一方、面積に関しては、演算に応じて VGLC の入力粒度を変えることで効率の良い実装が可能となり、その結果、 N_{LB} と W を抑えることができたと考えられる。これらのことは各プロセスに関して同様である。

ここで、実装面積とクリティカルパス遅延はトレードオフの関係を持つので、面積遅延積を導入する。図 7、図 8 に、乗算器と FFT の面積遅延積を示す。どちらの回路においても、提案論理セルが最も低い面積遅延積値になっており、特に FFT は MAC 演算で構成されていることから、その傾向が顕著であることがわかる。今回、

VGLC はスタンダードセルライブラリを用いて構成しているため、トランジスタレベルでの回路最適化を行うことにより、遅延、面積共に改善が期待できる。

5. まとめ

本稿では、入力粒度を可変にできる論理セルに対し、実際にベンチマーク回路を配置配線することで、面積及び遅延評価を行った。その結果、 $N=1, 4, 8$ の LC と比較して、VGLC の実装面積は最小となった。速度的に改良の余地はあるものの、面積遅延積も最小を示すことにより、VGLC は演算に応じて、回路を効率良く実装できていると考えられる。

今後は、配線構造の詳細設計も含めて、提案アーキテクチャのフィジカル設計を行う予定である。また、様々なアプリケーションを VGLC に効率よくマッピングするための CAE ツールの開発を行う。

参考文献

- [1] T. Sueyoshi and M. Iida, "Configurable and reconfigurable computing for digital signal processing," IEICE TRANS., vol. E85-A, no. 3, pp. 591-599, Mar. 2002.
- [2] 尼崎太樹, 中山英明, 濱邊直人, 飯田全広, 末吉敏則 "粒度可変構造を持つ再構成論理セルアーキテクチャの提案," 信学技報, RECONF2005-53, vol. 105, No. 450, pp.1-6, Nov. 2005.
- [3] 濱邊直人, 中山英明, 尼崎太樹, 飯田全広, 末吉敏則 "粒度可変構造を持つ再構成論理セルを用いた基本演算回路の実装," 信学技報, RECONF2005-54, vol. 105, No. 450, pp.7-12, Nov. 2005.
- [4] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.
- [5] "Opencores" [Online]. Available: <http://www.opencores.org/>
- [6] V. Betz, *VPR and T-VPack User's Manual (Version 4.30)*, 2000.
- [7] J. Rose, R. Francis, D. Lewis, and P. Chow, "Architecture of field-programmable gate arrays: The effect of logic block functionality on area efficiency," *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1217-1225, Oct. 1990.
- [8] A. Dehon, "Reconfigurable architectures for general-purpose computing," in *MIT, Artificial Intelligence Laboratory, AI Technical Report No.1586*, 1996.
- [9] *Xilinx XC4000 Data Sheet*, Xilinx, Inc., 1999.
- [10] "Berkeley Predictive Technology Model (BPTM)," [Online]. Available: <http://www-device.eecs.berkeley.edu/~ptm/interconnect.html>