

C-010

分散型デコーダを用いた命令デコーダの構成
Composition of the instruction decoder using a distributed decoder

土屋秀和† 八幡雅也†† 浅川毅†, ††
Hidekazu Tsuchiya† Masaya Yahata†† Takeshi Asakawa†, ††

1. まえがき

近年における FPGA や CPLD などの再構成可能なデバイスの普及に伴って、リコンフィギャラブル技術が注目されている。リコンフィギャラブル技術はプロセッサや DSP の回路に対して変更容易性の面より、エラーの修正や機能の拡張性などで期待されている[1]。リコンフィギャラブル技術に関し、我々は分散型デコーダを提案している[2]。本論では、分散型デコーダを教育用のプロセッサの命令デコーダに適用し、変更容易性、ハードウェア量、遅延時間の面より、評価を行った。

2. 教育用 8 ビットプロセッサ TAC1

教育用 8 ビットプロセッサ TAC1 は、学部生を対象とするハードウェア設計教育用として教材開発したものである。図 1 に TAC1 の基本構成を示す。命令処理部の構成に自由度を持たすため、プログラムメモリとデータメモリが分離されるハーバードアーキテクチャを採用している。プログラムメモリに格納される 14 ビット幅の命令は、7 ビットのエネーションコードと 7 ビットのエネラントに展開される。命令デコーダでは、7 ビットのエネーションコードを受けて、ALU やデータメモリなどの各構成要素に対して 10 ビットの制御信号を発生する。本研究では、TAC1 の命令デコーダを、分散型デコーダおよび組み合わせ回路によって構成し、比較評価を行った。

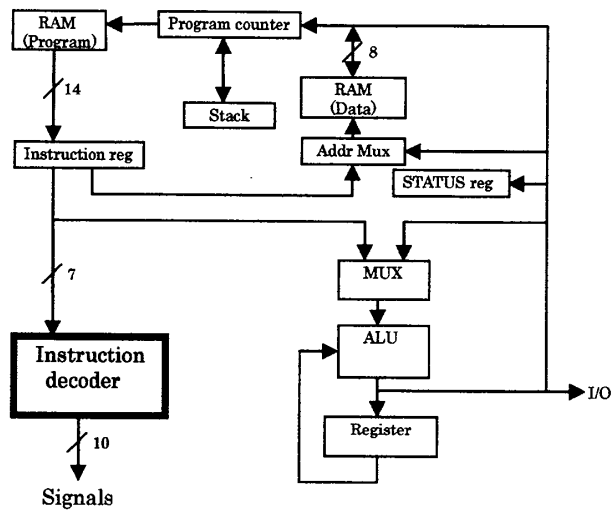


図 1: 教育用プロセッサ TAC1 の構成

† 東海大学工学研究科情報通信制御システム工学専攻
†† 東海大学電子情報学部コンピュータ応用工学科

表 1: 命令の構成と制御信号

命令	オペコード 7ビット		制御信号 10ビット							
			ALU	データ メモリ		MUX	REG	スタック 及びPC		
				W	R					
ADDW	00	0111	0	0100	0	0	1	1	0	0
ADDF	00	0111	1	0100	1	0	1	0	0	0
⋮										
XORF	00	0110	1	1110	1	1	1	0	0	0

3. 組み合わせ回路による命令デコーダの構成

図 2 に組み合わせ回路による命令デコーダの構成を示す。この回路は 7 ビットのエネーションコードを入力し、10 ビットの制御信号をデコード出力する。教育用プロセッサ TAC1 の基本 30 命令が入力状態の全組み合わせ $2^7=128$ 中に割り当てられ、残りの 98 通りはドントケアとなる。基本命令 30 に拡張命令を加える場合は、このドントケア部の入力条件を使用することにより、命令を拡張することは可能であるが、デコーダ全体を再構築する必要がある。

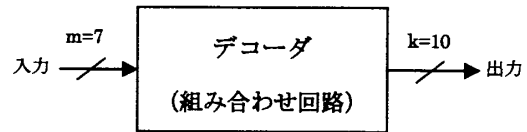


図 2: 組み合わせ回路による命令デコーダの構成

4. 分散型デコーダによる命令デコーダの構成

図 3 に分散型デコーダを用いた命令デコーダの構成を示す。分散型デコーダは要素デコーダを並列に組み合わせることによって構成される。各要素デコーダは割り当てられた条件に従って出力を活性化し、デコード出力する。そのため、教育用プロセッサ TAC1 の基本 30 命令に対して 30 個の要素デコーダを必要とする。命令の追加や変更に対して、要素デコーダを付加することで実現でき、拡張性および変更容易性に優れている。また、命令デコーダを稼働させたまま、要素デコーダを追加することができるため、動的リコンフィギャラブルへの応用が期待できる。

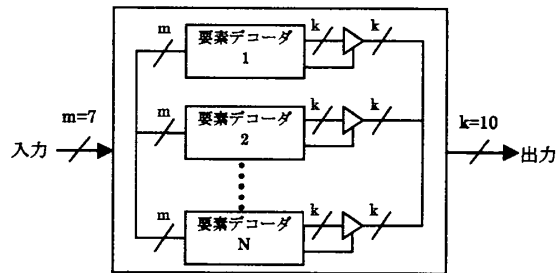


図 3: 分散型デコーダによる命令デコーダの構成

5. 実験

教育用プロセッサ TAC1 の命令デコーダを分散型デコーダによる手法と組み合わせ回路による手法で構成し、ハードウェア量とクリティカルパスの遅延時間を評価した。回路構成をVHDLで記述し、Xilinx社のIse7.1i[3]のデフォルト(スピード優先)条件で論理合成を行った。ハードウェア量は論理合成回路のゲート数で換算し、遅延時間はFPGA(Spartan III s50vqg100-4)デバイスに実装した時のクリティカルパスに対してシミュレーション値で求めた。

これら2つの構成手法を用いて、命令数30から50までの命令デコーダを構成し、その必要ゲート数及びゲート比率を表2に求め、図4に示した。評価した範囲では分散型デコーダ手法による構成のほうが組み合わせ回路構成に比べてハードウェア量が大きく、平均で1.28倍であった。

表2: 組み合わせ回路および分散型デコーダ手法の命令数に対する必要ゲート数

命令数	30	35	40	45	50
組み合わせ回路手法によるゲート数	282	336	507	636	651
分散型デコーダ手法によるゲート数	390	516	606	747	723
ゲート比率 [%]※	138.2	153.6	119.5	117.5	111.1

※ ゲート比率 = $\frac{\text{分散型デコーダ手法によるゲート数}}{\text{組み合わせ回路手法によるゲート数}} \times 100 [\%]$

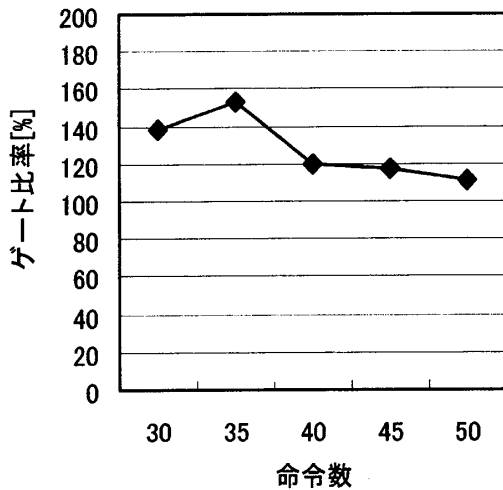


図4: 命令数に対するゲート比率

論理合成された回路に対してクリティカルパスにおける遅延時間と遅延時間比率を表3に求め、図5に示す。遅延時間は組み合わせ回路及び分散型デコーダを用いた手法ともに同程度であり大きな差は見られなかった。

表3: 組み合わせ回路および分散型デコーダ手法のクリティカルパスにおける遅延時間[ns]

命令数	30	35	40	45	50
組み合わせ回路手法による遅延時間[ns]	14.6	17.2	15.8	17.2	16.7
分散型デコーダ手法による遅延時間[ns]	16.6	16.8	15.2	18.6	16.4
遅延時間比率 [%]※	113.7	97.7	96.2	108.1	98.2

※ 遅延時間比率 = $\frac{\text{分散型デコーダ手法による遅延時間}}{\text{組み合わせ回路手法による遅延時間}} \times 100 [\%]$

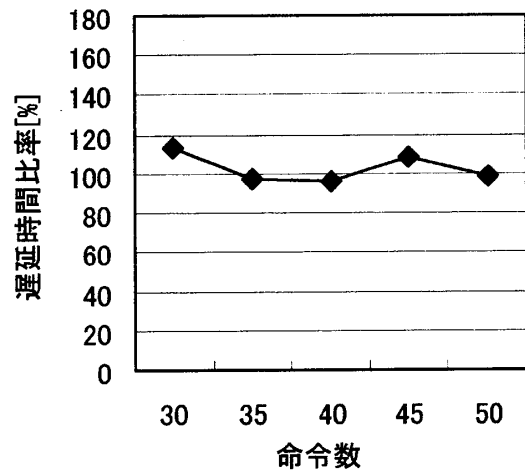


図5: 命令数に対する遅延時間比率

6. むすび

本研究ではリコンフィギャラブル指向の分散型デコーダを用いて、教育用プロセッサの命令デコーダを構成した。組み合わせ回路による構成と比較し、ハードウェア量と遅延時間を評価した。その結果、変更容易面では優位である分散型デコーダ構成であるが、ハードウェア量は組み合わせ回路手法と比べて、約1.3倍必要であることがわかった。遅延時間の面では同等の結果が得られた。今後の課題として、論理合成の条件を変更した際の影響、大規模な命令デコーダにおける評価、FPGAへの実装評価が挙げられる。

文献

- [1] アイピーフレックス:動的再構成でFFT 高速化高頻度切り替えの威力実証,日経エレクトロニクス,no.902,pp40-41,6-20,2005.
- [2] 堀桂太郎, 浅川毅:リコンフィギャラブル指向分散型デコーダに関する一検討,2004 ソサイエティ大会講演論文集,pp.84,2004.
- [3] Xilinx inc.:ISE7.1i, <http://www.xilinx.co.jp/ise/marketing/>, 2005.