

小規模かつ高性能を目指した 8bitSoC の LSI テスタを用いた解析

内田 裕之[†] 富川 大志[†] 山田 明宏^{††} 弘中 哲夫^{†††} 五島 敬史郎[†]
[†] 愛知工業大学 ^{††} ALSI デザイン ^{†††} 広島市立大学

1. はじめに

安価な小規模デバイスで SoC を構成するためには、回路規模が小さく、かつ、高性能なマイクロプロセッサが必要となる。そこで、数百ゲート規模で実装でき、柔軟なプログラミングが可能なマイクロプロセッサ実装を可能にする、Pilaf プロセッサの開発が行われている[1]。本研究では、LSI テスタを用いて Pilaf プロセッサを実装した 8bitSoC の解析を行う。

2. Pilaf プロセッサ

Pilaf プロセッサのブロック図を図 1 に示す。

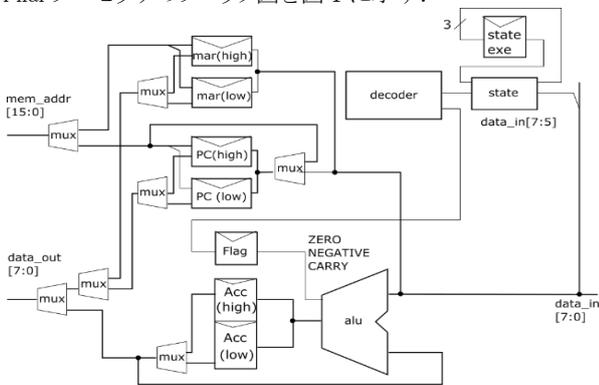


図 1 ブロック図

Pilaf プロセッサに実装しているレジスタは、Acc, PC, Memory Register, Flag の 4 つである。Memory Register はプロセッサ内部のレジスタを最小限に抑え小規模化するため、32 本のメモリレジスタを外部メモリ上に実装している。

3. 命令一覧

Pilaf プロセッサの命令一覧を表 1 に示す。誌面の都合上、特徴的な命令について解説する。ADDD 命令は、\$mX に格納されているアドレスのデータを用いて Acc と加算する命令である。STAD 命令は、\$mX に格納されているアドレスの番地に Acc の値を格納する命令である。JL 命令は、オペコードの後ろの引数アドレスに戻り値を格納し、condition(cond)と現在のフラグの論理積をとり、真なら cond の後の引数アドレスに格納してある飛び先アドレスをプログラムカウンタ(PC)に保存し、偽なら次の命令をフェッチする。

表 1 命令一覧

	オペコード	タイプ	命令操作	ZNC フラグ
1	NOR	F	\$Acc = \$Acc NOR \$mX	Z N 0
2	ADD	F	\$Acc = \$Acc + \$mX	Z N C
3	ADDD	F	\$Acc = \$Acc + Mem[\$mX]	Z N C
4	MOVE	F	\$mX = \$Acc	---
5	STAD	F	Mem[\$mX] = \$Acc	---
6	SETI	I	\$Acc = immediate	Z N 0
7	JL	J	if(flag & cond) { \$mX = PC PC = \$mX }	---
8	SRL	F	\$Acc = \$Acc >> 1	Z N 0

4. Personal Ocelot

Personal Ocelot は今回の解析で用いた LSI テスタであり、テスト対象デバイスの構造上のエラーの検出や LSI 内部の素子間の動作速度の計測を行うために設計されたテスタ機である。今回の解析では、テスト対象デバイスの内部回路における不良や論理機能が正常か否かの判定を行った後、図 2 のように、基準信号から出力信号がどれほど遅延しているかを観測し、最短動作速度を解析する。

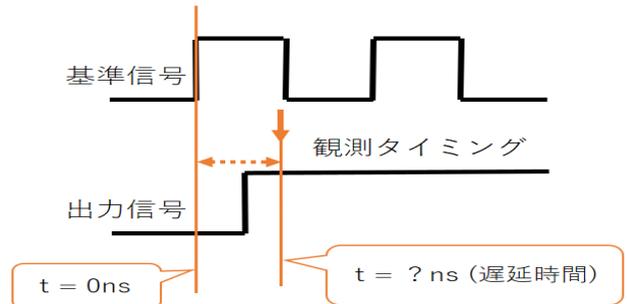


図 2 動作速度の観測

5. 結果とまとめ

本稿では、Pilaf プロセッサのテスト結果について検討する。図 3 は Pilaf プロセッサの動作の一例を示す。Pilaf プロセッサは正常に動作し、最も遅延している出力端子は、clk の立ち上がりを基準 (t=0)として、約 512ns 遅れて立ち上がる事が観測できた。遅延時間を逆数にして計算をすると、最大動作周波数は 1.95MHz と推定できる。今後は SRAM も含めた Pilaf SoC の動作速度について詳細に解析していく予定である。

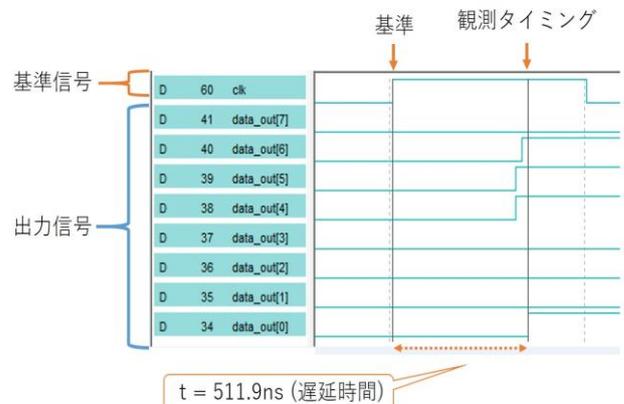


図 3 Pilaf プロセッサのテスト結果

参考文献

[1] 米田 浩貴, 胡濱 良樹, 谷川 一樹, 児島 彰, 弘中 哲夫: "Pilaf:超小型プロセッサ IP の開発", 電子・情報・システム部門 電子回路研究会, ECT-15-71, (2015)