

ASIC 設計手法を用いた LSI 回路設計に関する研究

Study on LSI Circuit Design using Application Specific Integrated Circuit Technology.

鈴木貴斗 †, 五島敬史郎 † †, 江口一彦 † †
Takato Suzuki, Keishiro Gosima, Kazuhiko Eguchi

Abstract: HDL (Hardware Description Language) is the most powerful tools for the design and development of Large Integrate Circuit (LSI). In particularly, the designing of FPGA (Field Programmable Gate Array) is increased because of reconfigurable device. However, FPGA has two demerits for large consumption and low processing capability. In this paper, we utilized to LSI design with ASIC (Application Specific Integrated Circuit) technology because of low consumption and short processing time. Using LSI design development tools of VDEC (VLSI Design and Education Center) based on the license, ASIC development environment was almost prepared. We designed, fabricated and measured for three LSIs by means of ASIC technology.

1. はじめに

昨今、身の回りにはコンピュータ製品があふれ設計技術者の重要性がますます高まってきている。さらに、デジタル回路を用いた大規模集積回路設計の現場では、設計の効率化のため、HDL (Hardware Description Language) と呼ばれるハードウェア言語の利用が主として行われており、現在は C 言語での開発も可能となってきている。大学では、デジタル回路・学生実験等の講義で、簡単な組み合わせ回路や順序回路について学習し学生実験においては、FPGA (Field Programmable Gate Array) を用いた実習を行っているが、デジタル回路の動作の仕組みと、高度に抽象化された HDL 言語を用いたデジタル回路動作との間には、大きな理解の障壁があるのも事実である。つまり、FPGA を用いたデジタル回路設計において FPGA チップ内部はブラックボックス化され、内部でどのような回路構成になっているかを確かめることも難しい。加えて、高機能・高性能化によって半導体集積回路は、ますます大規模および高い動作周波数を要求されている。特に高い動作周波数の動作状況下では、デジタ

ル回路であってもアナログの設計要素が必須になってきている¹⁾。開発・設計の人材供給を求められる本学にとっては、HDL を用いた開発環境だけではなくその裏で動いている回路の中身や動作を知っておく必要があると考える。

そこで、我々は、HDL によるデジタル回路設計だけでなく、ASIC (Application Specific Integrated Circuit) の設計まで拡張した研究を行った。ASIC 開発では、HDL を用いながらデジタル回路の構成要素である CMOS トランジスタの設計から配置、それに伴う信号遅延等を考慮しなくてはならず、内部回路が理解できていないと設計は難しい。²⁾

我々は、ASIC 設計・開発環境を共同研究を通じて整備した。構築した設計開発環境は、東京大学大規模集積システム設計教育研究センター (VDEC VLSI Design and Education Center) が高等教育機関向けに提供しているライセンスを用いて設計・開発ツールを導入したものである。

2. ASIC 開発環境

今回は、まずデジタル集積回路の設計環境を整備した。現在のデジタル回路 LSI の設計は EDA による自動

† 愛知工業大学大学院 電気電子工学専攻 (豊田市)

† † 愛知工業大学 工学部 電気工学科 (豊田市)

化が進んでいるため、CAD ツール使用が設計には必須である。VDEC から EDA ツールを導入し整備した。

① 論理シミュレーション

Icarus Verilog

② 回路シミュレーション

Synopsys 社 HSPICE、nanosim

③ 論理合成

Synopsys 社 Design Compiler

④ 自動配置配線ツール

Synopsys 社 IC Compiler

⑤ レイアウト設計

JDAT 社 ISMO, ASCA

⑥ LSI チップ製造

フェニテックセミコンダクター (株)

①~⑤までの工程は、VDEC 環境において整備を行い本学で実施可能になった。また⑥のチップ製作工程では、フェニテックセミコンダクター社の行っている大学等の研究・教育を目的としたシャトル便制度をもちいて製作を依頼した。

3. インバータの設計

デジタル回路における最も基本回路が NOT 回路 (インバータ回路) である。入力信号を反転して出力させる回路であるが、この回路を実現するためには図 3.1 で示すように PMOS 型・NMOS 型の電界効果型トランジスタが必要になる。図 3.2 は ASIC としてシリコン基板上に実際に NOT 回路として作製した図である。

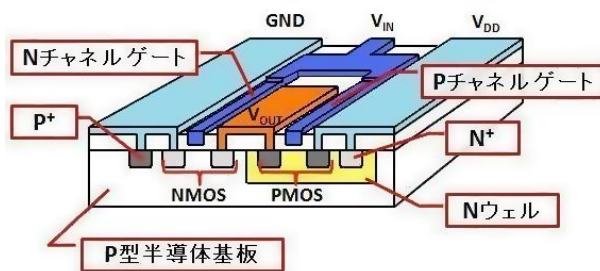


図 3.2 インバータ回路の断面構造

P 型半導体の基板に組み込んだ形で NMOS と PMOS を作り、その上に電源端子(VDD)や GND、入力端子(VIN)、出力端子(VOUT)への配線を被せている。この上に被せていく構造を階層構造という。各階層のトランジスタ配置には設計ルールがありチャンネルゲート印加電圧やチャンネルゲート幅などの設定によって動作スピードや耐ノイズ性能などを決めることが出来る。今回は、製造側の推奨値を用いて設計を行っている。

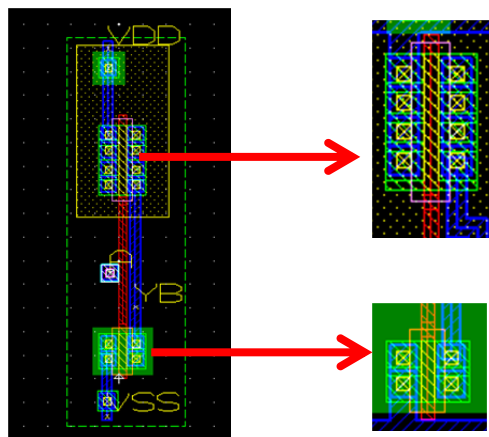


図 3.3 インバータ回路のレイアウト

図 3.3 は、トランジスタの各階層設計ルールに基づきレイアウト設計を行ったものである。上半分が PMOS トランジスタ、下半分が NMOS トランジスタの設計を行っている。PMOS では正孔・NMOS では電子が支配的に働く為シリコン中の有効質量を考慮した設計が行われている。このように、半導体物性の効果も設計要素として取り入れなければ ASIC 設計は不可能である。実際の設計では JDAT 社の CAD ソフトを用いて設計するが各要素の動作や意味の理解無しにレイアウト設計は出来ない。(3),(4)

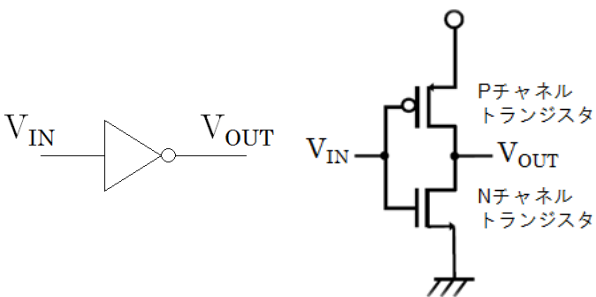


図 3.1 インバータ回路の論理記号とトランジスタ回路

4. 7セグメント LED 制御用 LSI の設計

7セグメントディスプレイ (seven-segment display) は、電子的な表示装置の一種であり、十進のアラビア数字を表示することができる。(図 3.1)アラビア数字一文字を表現するために、それぞれ個別に点灯・消灯できる7つのセグメントから構成されているためこの名がある。我々は、この制御用 LSI の設計を行った。

4.1 設計仕様

LSI の入力端子は DPNT, NUM3, NUM2, NUM1, NUM0 の5つがあり、この5つに2進数で0~15までの数値を入力していく。順番に5桁目が DPNT, 4桁目が NUM3, 3桁目が NUM2, 2桁目が NUM1, 1桁目が NUM0 と入力されるよう FPGA での PIN 配置を行う。この入力端子にそれぞれ2進数で信号が与えられることで、LSI 内部で7セグメントディスプレイを点灯させるための操作が行われる。出力端子は7セグメントディスプレイのセグメント点灯端子に相当する8つが設けられており、それぞれ SEGDP, SEGA, SEGB, SEGC, SEGD, SEGE, SEGF, SEGG としている。(4)

4.2 RTL 設計

設計仕様に基づいて、HDL ハードウェア言語を用いて RTL 設計を行う。入力は、4桁の2進数と小数点を含めた5ケタで00000~11111までの入力とする。出力は、10進数表記で0~9、及びA~Fまでを表示する。今回 HDL 言語は Verilog を用いた。

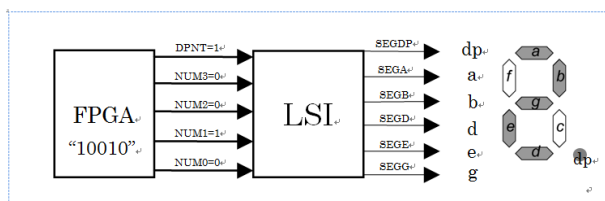


図 4.1 7セグメント制御 LSI の入出力

図 4.2 は、7セグメント LED 制御用 LSI の RTL 設計か

ら Synopsys 社の Design Compiler によって論理合成を行いデジタル論理回路で表記したものである。デジタル回路の基本素子である NOT・AND・OR・NOR・NAND・XOR 回路から構成されている事が分かる。

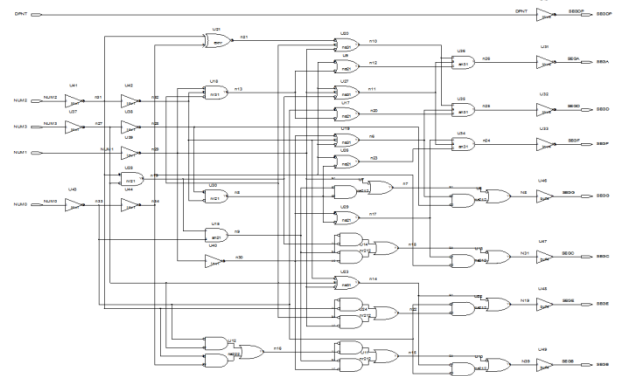


図 4.2 7セグメント制御 LSI の論理回路

4.3 レイアウト設計

図 4.2 を基にレイアウト設計を行った回路図を図 4.3・4.4 に示す。レイアウトの条件は基本素子 (NOT・AND・OR・NOR・NAND・XOR) の各サイズは共通で統一した。論理回路図も同じものを使用している。しかし、2つの図を比較すると全くレイアウトが違う。図 4.3 は、耐ノイズ性能を高める為に各素子間に隙間を設けて隣り合う

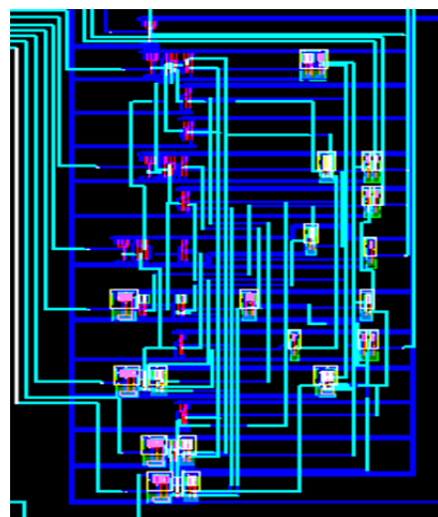


図 4.3 耐ノイズを考慮したレイアウト

信号線間のノイズの影響を受けにくいように設計したものである。図 4.4 は、高速動作を目的としたレイアウト設計である。この場合は、素子間の信号遅延時間を最小限にするため、最少面積で設定したものである。

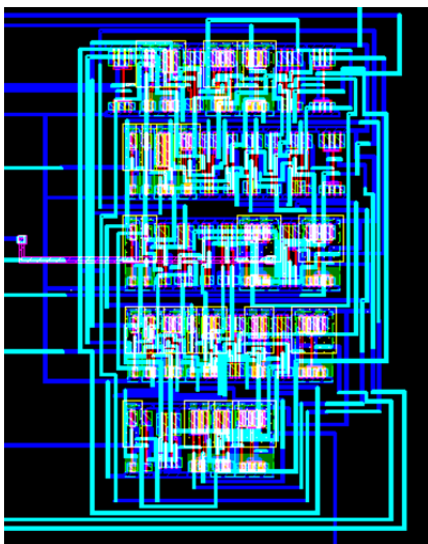


図 4.4 高速動作を考慮したレイアウト

この様に、レイアウト設計でも完成された LSI の性能は大きく変わることが実証された。このほかにも配線の引き回しによっても回路の性能は変化することが確かめられた。

5. ASIC 手法を用いた CPU の設計

アーキテクチャ設計では主に設計対象の仕様の決定をおこなう。機能と端子数、外部接続の内容や扱う命令コードなどを細かく決定する。

今回の設計では、LSI は 4bit CPU とした。2 進数 8bit の命令信号を取り扱い、4bit の算術演算、bit 操作、論理演算が可能である。また算術演算において桁上りを認識する。一例として、命令分類と全体のブロック図を示す。(5),(6)

命令分類	命令
算術演算命令	ADD / SUB
論理演算命令	AND / OR / EXOR
ビット操作命令	ROL / ROR
ロード命令	LDA
ストア命令	STA
分岐命令	BRZ / BRC / JMP
その他	NOP

図 5.1 今回設計 CPU の命令セット

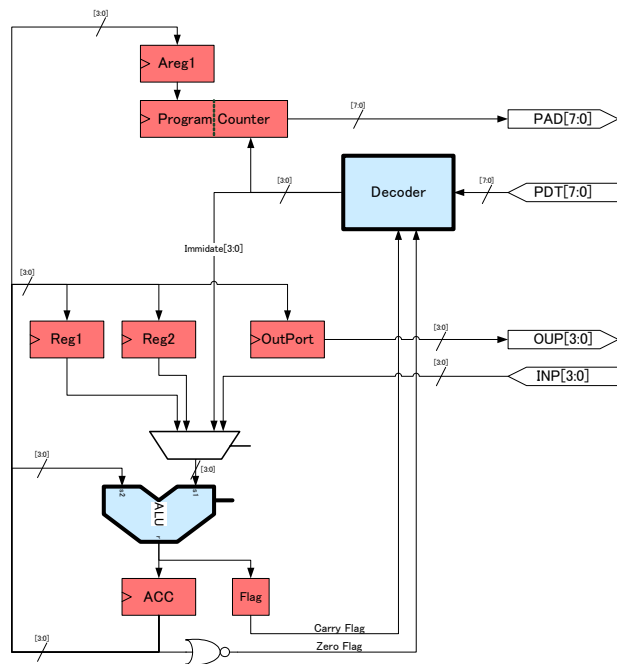


図 5.2 CPU の構成図

CPU は大きく分けて 5 つのユニットから構成される。プログラムカウンタ回路・デコーダ回路・レジスタ回路・ALU (算術演算回路)・入出力制御回路 である。

ここでは、一例として ALU の動作について述べる。ALU にはメモリとアキュムレータより入力された 2 値のデータ(ここでは 4bit とする)を制御信号の内容に応じて演算を行う。得られた演算結果はラッチ(latch)へ転送され、その後、アキュムレータに保存される。また、演算の際に結果がゼロになったか、桁あふれが生じたかどうかを示す情報として、ゼロフラグとキャリーフラグとしてそれぞれ記憶する。図 5.3 と 5.4 に ALU の回路構成を記す。

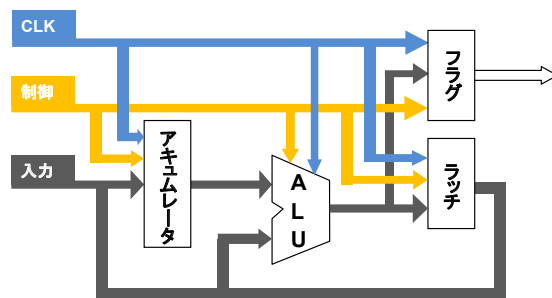


図 5.3 ALU の回路構成

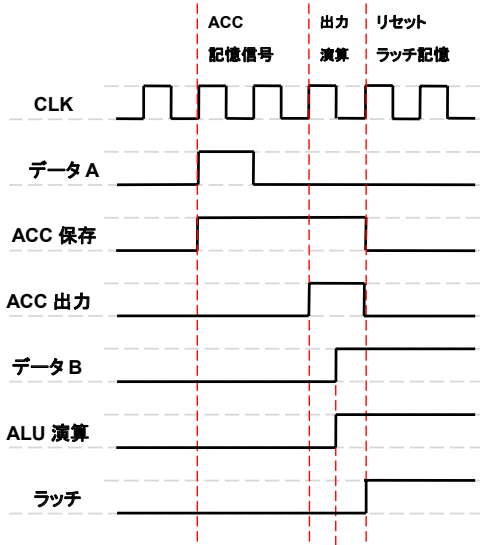


図 5.4 ALU の動作タイミング

次に ALU の動作の仕組みについて、動作波形例を用いて記す。

1. 制御信号は CLK(クロック)の立上げに合わせて送られ、各所に制御命令を与える。
2. メモリなどの記憶領域から出力されるデータ A を ACC に加えられる記憶せよ、との信号を受け取ることで ACC に記憶する。
3. ACC に記憶データを出力、ALU に演算せよ、との制御信号が入ることによって ACC は記憶されているデータを出力し、ALU は演算のスタンバイに入る。
4. 他記憶領域(ここでは ACC 以外とする)からデータ B が ALU に向けて出力される。ALU は既に演算のスタンバイがされており、また、ALU は CLK の信号を受けていないため、CLK に同期しては動かない。そのため、データ A と B の 2 値が揃って入力された時点で演算を行い、演算結果を出力する。
5. ALU より出力されたデータはそれぞれ flag と latch へ記憶される。flag と latch は CLK 信号を受けているので、CLK に同期して演算結果を記憶する。
6. latch から ACC への入出力もそれぞれに出力と入力制御信号が入られることでおこなわれる。

Verilog HDL 記述を用いて論理合成した論理回路図を図 5.5 に示す。

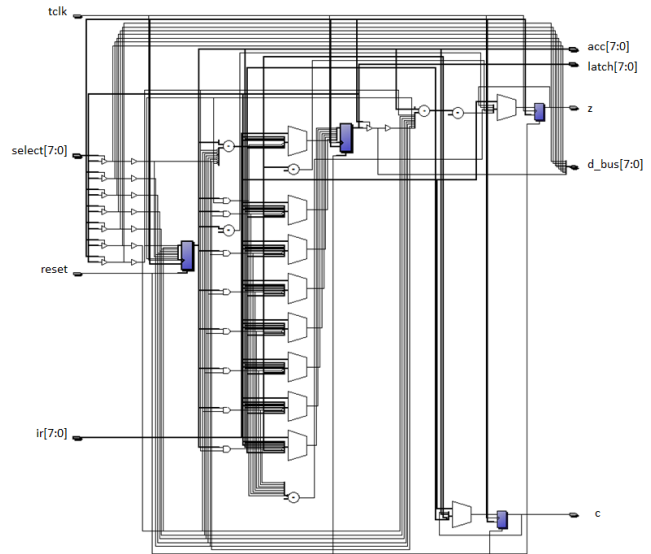


図 5.5 ALU の論理回路図

5.1 配置配線ツールを用いたレイアウト設計

CPU のような大規模デジタル回路になると、使用するトランジスタの数も約 5000 個にも達する為、個々のトランジスタを手作業で配線することは非常に難しい。そこで配置配線ツールを用いてレイアウトを行った。使用ツールは、Synopsys 社の IC Compiler である。

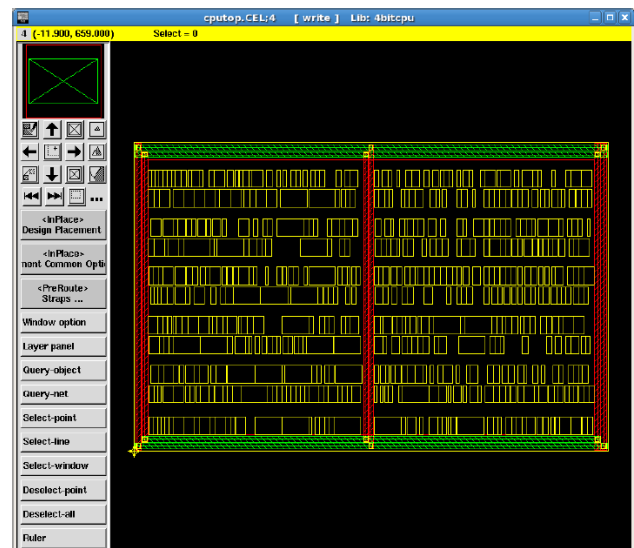


図 5.6 IC Compiler を用いた配置配線

今回、レイアウトでの条件は電源ラインの強化によって生じる誤作動の電圧レベル差を重点に行った。レイアウトでは、図 5.6 に示すように長方形の枠中に収める

ように設計するが、今回は枠の内側を囲むように電源と GND ラインを配置した。さらに中央部分に電源ラインを配置したレイアウトと、電源の強化を目指し、左右 2 カ所に電源ラインを挿入した 2 つの場合を想定した

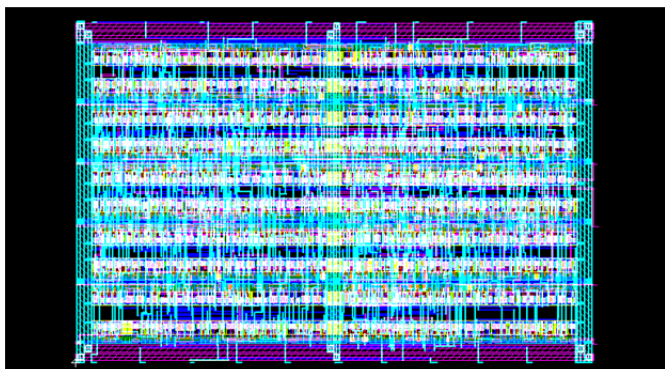


図 5.7 中央部に電源ライン配置したレイアウト

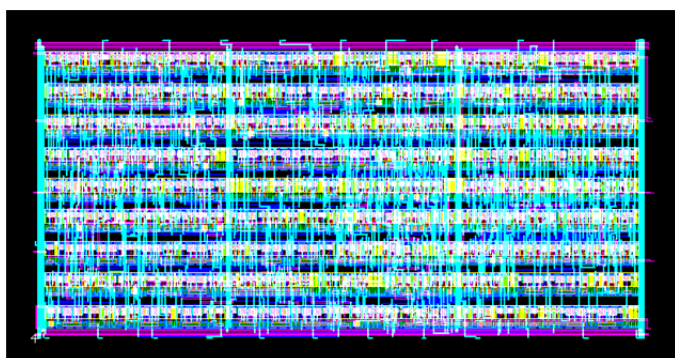


図 5.8 2カ所に電源ライン配置したレイアウト

図 5.7 と図 5.8 は両方共 4bitCPU の CMOS 回路である。配線は第 1 層が青色、第 2 層が水色、第 3 層が紫色の表示となっている。赤色の破線枠で囲まれた部分は電源補強ラインである。

6. 性能評価

本研究の特色の一つは、設計した回路をシュミレーションによる確認だけではなく、実際の LSI チップとして製作し、当初予想した性能を満たしているかを評価する点にある。

今回の成果としては、インバータ回路及び 7 セグメント LED 制御 LSI については、FPGA などを用いたテスト機

を自作して動作確認・設計通りの性能を確認することが出来た。

しかし、CPU については、テスト項目が多岐にわたる為、動作確認は行っていない。LSI テスタと呼ばれる専用機を用いて実施することを検討している。

7. まとめ

本研究では、ASIC 設計手法を軸としたデジタル回路設計における研究を行った。ASIC 開発環境は、東京大学大規模集積システム設計教育研究センター (VDEC) と共同研究を通じて整備を行い、チップ製作にはフェニテックセミコンダクター社の協力を得た。

この研究を通じて、デジタル回路の設計であっても HDL ハードウェア言語による設計だけでなく、デジタル回路の構成要素である NMOS・PMOS トランジスタ 1 個の NOT 回路動作理解から配置、ノイズに強い設計方法やそれに伴う信号遅延等を考慮した設計を行うことができた。また、CPU 設計においては、アーキテクチャ設計から初めて実際の回路として LSI チップ製作まで行うことが出来た。これらの設計手法は HDL を用いた FPGA 設計では考慮できない ASIC の高度な設計を行うことが出来た。

今後の展開として、昨年本研究室で作成した CPU をベースとし、パイプライン処理の機能の追加を行う。これにより各命令の実行時間は変わらないが、スループットが向上するため全体的な処理の実行が速くなる。各命令は実行順に、命令フェッチ(以下 IF)、命令デコード(以下 ID)、命令実行(以下 EX)、メモリアクセス(以下 MA)、ライトバック(WB)と 5 つの処理に分けられる。以下にパイプライン処理と逐次処理の時間遷移と実行処理との関係を示す。

クロック	1	2	3	4	5	6	7	8	9	10
命令1	IF	ID	EX	MEM	WB					
命令2		IF	ID	EX	MEM	WB				

図 7.1 パイプライン処理の時間遷移と実行処理

クロック	1	2	3	4	5	6	7	8	9	10
命令1	IF	ID	EX	MEM	WB					
命令2						IF	ID	EX	MEM	WB

図 7.2 逐次処理の時間遷移と実行処理

謝辞

本研究は、愛知工業大学 H25 年度特別研究助成、ALSI デザイン(株)、フェニテックセミコンダクター(株)、及び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じ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) 松尾 和典 他
熊本電波高専 研究紀要 第 34 号 11p
(2007)
 - 2) VDEC 監修 : デジタル集積回路の設計と試作
培風館 (2001)
 - 3) タウア・ニン : 最新 VLSI の基礎 第 2 版
丸善出版 (2013)
 - 4) Behzad Razavi : アナログ CMOS 集積回路
の設計 丸善出版(2013)
 - 5) パターソン&ヘネシー : コンピュータの構
成と設計 日経 BP 出版 (2013)
 - 6) 岩出秀平 清水徹 : 実用プロセッサ技術
ムイスリ出版 (2012)
(受理 平成 26 年 3 月 19 日)