

## 計算機工学一貫教育用マイクロプロセッサ QP-DLX の開発

岩井原 瑞穂 中村 秀一 國貞 勝弘 斎藤 靖彦 中川 智水 永浦 渉  
山家 陽 山田 哲也 赤星 博輝 權 五鳳 村上 和彰 安浦 寛人

九州大学 大学院総合理工学研究科 情報システム学専攻

〒 816 春日市春日公園 6-1

E-mail: soshiki@is.kyushu-u.ac.jp

あらまし

【ソフトウェアから、計算機アーキテクチャ、論理回路設計、集積回路設計・製造に至る計算機工学一貫教育】用マイクロプロセッサ QP-DLX(Kyushu University Education-Purpose DLX Microprocessor) の開発プロジェクトについて述べる。アーキテクチャとしては、「J. L. Hennessy and D. A. Patterson: *Computer Architecture: A Quantitative Approach*」の中で教材として用いられている DLX アーキテクチャを採用している。QP-DLX の開発構想を述べ、命令パイプライン、内部動作の観測機能、割込みなどの仕様、および開発状況について報告する。

和文キーワード：教育用マイクロプロセッサ, DLX, RISC, 命令セット・アーキテクチャ, パイプライン処理, 割込み

## Developing an Education-Purpose Microprocessor QP-DLX for Computer Science Courses

Mizuho IWAIHARA, Shuichi NAKAMURA, Katuhiro KUNISADA, Yasuhiko SAITOH,  
Tomomi NAKAGAWA, Wataru NAGAURA, Akira YAMAGA, Tetsuya YAMADA,  
Hiroki AKABOSHI, Oubong GWUN, Kazuaki MURAKAMI and Hiroto YASUURA

Department of Information Systems  
Interdisciplinary Graduate School of Engineering Sciences  
Kyushu University  
Kasuga-shi, Fukuoka 816 Japan

E-mail: soshiki@is.kyushu-u.ac.jp

Abstract

This paper reports on the development project of the education-purpose microprocessor QP-DLX (Kyushu University Education-Purpose DLX Microprocessor) for computer science courses. The ISP architecture and the basic pipeline structure of QP-DLX is based on the DLX, which is used in J. L. Hennessy and D. A. Patterson: *Computer Architecture: A Quantitative Approach*. This paper explains the objectives of the project, and gives some architectural specifications of QP-DLX, such as pipeline design, observation function, and interrupt handling.

英文 keywords: education-purpose microprocessor, DLX, RISC, ISP architecture, pipelining, interrupt

## 1 まえがき

情報工学や計算機工学の教育において「ソフトウェアから、計算機アーキテクチャ、論理回路設計、集積回路設計・製造に至る計算機工学の一貫教育」の必要性はこれまでも各所で指摘されてきた。大学等における計算機工学教育の現状は、プログラミング教育を中心としたソフトウェア教育にカリキュラムの重点が置かれ、ハードウェア教育との均衡がとられているとは言い難い。また、ハードウェア教育だけを見ても、学生達は計算機アーキテクチャ、論理回路理論、集積回路に関する講義で得た知識を簡単な論理回路の製作や市販のマイクロプロセッサを用いた実験で確認するに留まり、アーキテクチャ、論理回路、電子回路の相互の関連をこれらの実験から実感することは難しい。またソフトウェア演習やハードウェア実験で教材として使用される計算機モデルについても、現在の技術水準からは程遠く、計算機に関する実践的かつ実務レベルで役立つ知識を得るための教材としては不十分である。

米国においては1980年代初頭から、システム設計の立場からの集積回路設計教育が進められており、MOSIS等の組織を利用して学生実験で集積回路製作が体験可能となっている。また、ヨーロッパにおいても、仏ES2社等の協力の下、仏IMAGを中心としたEUROCHIPプロジェクト等により、大学の教育研究目的で集積回路の作製が可能な状況が作られている。我が国でも、大学等の教育機関における集積回路設計・製作演習の必要性がようやく認識されており、教育機関のために集積回路を製造する組織の設立も具体化しつつあるが、現状において集積回路設計演習まで行っている大学はまだ極めて少ない。

しかし、高性能ワークステーションおよび論理回路CADの普及やハードウェア記述言語によるアーキテクチャ設計法の発展により、大学においても比較的短期間でマイクロプロセッサを設計することが可能になりつつあるため、今後大学等において計算機アーキテクチャ、論理設計、集積回路設計の教育や演習を統合して行うようになると思われる。

以上の状況に鑑み、我々は、コンパイラやOSなどソフトウェアから、計算機アーキテクチャ、論理回路設計、集積回路設計・製造に至るまでの一貫した教育の共通の教材として用いることができ、しかも計算機アーキテクチャの現在の技術水準を反映した「計算機工学の一貫教育」用マイクロプロセッサ開発プロジェクトを進めている[7][8]。本稿では、現在開発中であるマイクロプロセッサQP-DLX(Kyushu University Education-Purpose DLX Microprocessor)について報告する。まず、2章で、開発プロジェクト全体の方針を述べる。そして、3章で、アーキテクチャとして採択したDLXについてその仕様を与え、4章以降で、現在開発中のプロトタイプの主な仕様(命令パイプラインと割込み、観測機能)を紹介する。

## 2 開発方針

### 2.1 利用目的

①計算機アーキテクチャの講義、②当該計算機アーキテクチャに基づいたマイクロプロセッサの設計・製作、そして、③当該マイクロプロセッサ上でのソフトウェア(OS、コンパイラ、アプリケーション、等)の作成、を一貫して1つのアーキテクチャおよびマイクロプロセッサを用いて教育することの効果は絶大である。そこで、我々は、次の利用目的を想定して、「ソフトウェアから、計算機アーキテクチャ、論理回路設計、集積回路設計・製造に至る計算機工学一貫教育」用マイクロプロセッサを開発することにした。

### 2.2 利用目的

- 計算機アーキテクチャ講義用教材：本マイクロプロセッサを教材として用いて、現在の計算機アーキテクチャの技術水準(パイプライン処理、RISC、キャッシュ、仮想記憶、等)を反映した計算機アーキテクチャの講義が行えること。
- ソフトウェア演習用教材：本マイクロプロセッサをターゲット・マシンとしてプログラミング演習やシステム・プログラム(OS、コンパイラ、等)製作実習が行えること。
- ハードウェア実験用教材：本マイクロプロセッサの仕様を基に、論理回路設計実習、さらに、ハードウェア製作が可能なこと。さらに、CAD等の設備が揃っていれば、集積回路の設計・製作実習が可能なこと。
- ハードウェア実験用パーツ：上記のようなハードウェア製作が設備上不可能な場合、本マイクロプロセッサ自身をパーツとして用いて、設計した論理回路を実装可能なこと。すなわち、一種のFPD(Field Programmable Device)として使用可能なこと。

### 2.3 教育用マイクロプロセッサの現状

既に述べたように、現時点では、1つのアーキテクチャおよびマイクロプロセッサを教材として用いての「ソフトウェアから、計算機アーキテクチャ、論理回路設計、集積回路設計・製造に至る計算機工学一貫教育」は行われるに至っていない。以下に、ハードウェア実験、および、ソフトウェア演習、等で教材・パーツとして用いられているマイクロプロセッサの現状をまとめる。

- 市販マイクロプロセッサの利用：たとえば、Z80やM68000といった初期の市販マイクロプロセッサを用いて、アセンブリ・プログラミング演習やマイクロコンピュータ・システム製作を行う。
  - － 長所：安価である。マイクロプロセッサを搭載する基板の製作が容易である。
  - － 短所：アーキテクチャが単純ではないので、マイクロプロセッサそのものをハードウェア実験で設計・製作するのが難しい。そのためマイクロプロセッサの外部インタフェースのみの設計・製作で終わる場合が多い。計算機アーキテクチャの現在の技術水準から遅れている(ただし、現在の高性能RISCマイクロプロセッサは高価であり、また、その基板製作はかなり高度な技術が要求される)。
- 仮想マイクロプロセッサのソフトウェア・シミュレータ：仮想マイクロプロセッサのシミュレータを作成し、専らソフトウェア演習に用いる。
  - － 長所：教育を目的として設計しているので、本質的でない仕様にも煩わされない。環境さえ整っていれば、学生1人1人がシミュレータをほぼ無償で使用できる。
  - － 短所：名前の通り仮想マイクロプロセッサなので実体が存在せず、ハードウェア実験の教材としては利用できない。コンパイラやOSなどの演習において性能評価を行なう場合、シミュレータでは実行速度が問題となることがある。
- オリジナル実マイクロプロセッサ：教育用に実際にマイクロプロセッサを製作する。これには、たとえば、京都大学や京

都高度技術研究所が中心となって開発した「計算機工学・集積回路工学教育用マイクロプロセッサ」KUE-CHIP（1989年）[2]、KUE-CHIP2（1992年）[4]や、九州工業大学が開発した「再構成可能な論理LSIを用いた教育用マイクロプロセッサ」KITE（1992年）[5]がある。

- 長所：教育を目的として設計しているため、本質的でない仕様を含まない。加えて、ハードウェアの入門教育を目的として、マイクロプロセッサ内の内部状態の可観測性を重視している。極めて単純なアーキテクチャとなっているので、環境さえ整っていれば、実際にハードウェア実験で当該マイクロプロセッサそのものを設計・製作できる。
- 短所：語長やアドレス長の制限がきつ、かつ、割込みをサポートしていないので、ソフトウェア演習用教材としては使いにくい。計算機アーキテクチャの現在の技術水準（特に、パイプライン処理、キャッシュ、仮想記憶）が反映されていない。

## 2.4 QP-DLX プロジェクトの方針

- DLXアーキテクチャ：初期のハードウェア教育として計算機の原理を理解させるというよりはむしろ、学部専門課程あるいは修士課程を対象した高度な講義・演習のための教材とする。そのため単純、かつ、現在の技術水準（RISC、32ビット語長、パイプライン処理、割込み、キャッシュ、仮想記憶、等）を反映した計算機アーキテクチャとする。このため命令セット・アーキテクチャおよび命令パイプライン構造は、計算機アーキテクチャの教科書として米国で広く利用されている文献[6]のDLXに基づくことにした。
- CADシステムの利用：DLXアーキテクチャの採用に伴ない、マイクロプロセッサの規模が8ビットのときより増大し、学生実験において実際にTTL IC等で製作可能なハードウェア量ではなくなる。代りに、VLSI用CADを用いた設計を課題とする。近年、教育環境での高性能ワークステーションの導入が進み、高機能CADの導入も可能となってきている。またハードウェア記述言語や自動論理合成・自動レイアウト機能などの最新の技術を駆使することにより、QP-DLX程度のLSIの設計が学生演習で可能になりつつある。実際にこれらの技術を学生が体験してみることで、集積回路工学への理解が大きく進むであろう。
- フィールド・プログラマビリティ：演習で設計されたマイクロプロセッサを製造することは、教育機関に限られた予算と設備では困難を伴う。そのためプロセッサの一部をフィールド・プログラマブルとし、設計された論理回路の実装が行えるようにする。具体的には、①プロセッサ全体をFPGA (Field Programmable Gate Array) で実現する、②制御部のみをPPGAで実現する、③制御部をマイクロプログラム制御とする、といった方法を採用する。
- 観測機能：マイクロプロセッサ内部の信号をソフトウェアの介入なしに観測する機能、および1クロックサイクルごとに命令の実行/停止を行なう機能を備えることにする。従来のアーキテクチャの講義では内部動作は原理的な説明にとどまりがちであったのに対し、観測機能により実際に命令がパイプラインを流れる様子や、パイプラインで生じるハザードが解決される様子の観察が可能になり、学生のアーキテクチャ

動作原理の理解に役立つものと思われる。また論理設計においては、学生に大まかなプロセッサ仕様のみを与えておき、QP-DLXの観測を通して細部の制御論理を推測させる方法がとれる。さらにプロセッサの製品テストに観測機能を用いることにより、特別な測定機器なしでの検査が可能になる。

## 2.5 開発計画

本プロジェクトで順次開発してゆく予定のものを述べる。

- QP-DLX チップ：DLX仕様に割込みおよび観測機能を加えた32ビット・マイクロプロセッサ・チップを設計・製造する。しかし、前出のフィールド・プログラマビリティは備えず、QP-DLXはその前段階のプロトタイプという位置付けである。
- QP-DLXの設計データの公開：QP-DLXのCAD上での設計データをハードウェア記述言語やレイアウトなどの集積回路工学の教材として提供する。またQP-DLXの設計ドキュメントおよびソース・コードを広く公開し、論理合成や設計検証のベンチマークとして利用してもらう。
- QP-DLX システム・キット：ソフトウェア演習向けに、システム・ボードおよびシステム・ソフトウェア（モニタ、アセンブラ、コンパイラ、等）を開発する。
- QP-DLX プロダクト版：フィールド・プログラマビリティの機能を備えたプロダクトを設計・製作する。その際、上記プロトタイプから得られるハードウェア量、性能、設計に要する工数、製造コストなどのデータを参考に、ハードウェア実験の限られた時間内で可能な設計量を勘案して、フィールド・プログラマブルな部分を定める。実際に大学等の教育機関における教育用として、広く配布する。またハードウェア実験向けに、学生が設計した論理回路を本プロダクトにフィールド・プログラムするためのCADツールを開発・配布する。

## 3 アーキテクチャ仕様

DLXは、計算機アーキテクチャの教科書として米国で広く利用されているJ. L. Hennessy and D. A. Patterson: *Computer Architecture: A Quantitative Approach*[6]の中で教材として用いられているアーキテクチャである。DLXは、今日のアーキテクチャの主流であるRISCアーキテクチャに基づいており、RISC型商用マイクロプロセッサの多くを平均化したようなアーキテクチャとなっている。

QP-DLXはDLXのアーキテクチャを採用しており、これに観測機能などの新たな機能を追加している。また、回路規模等の制限により、浮動小数点関連命令および除算命令は実装していない。文献[6]では、DLXの基本仕様を定義するにとどめており、オペコード割当てなどの細部の仕様や割込み仕様は与えられていないため、これらの仕様を独自に定めている。以下、QP-DLXの仕様を述べる。

- 基本語長：32ビット
- 汎用レジスタ：32ビット汎用レジスタR0-R31

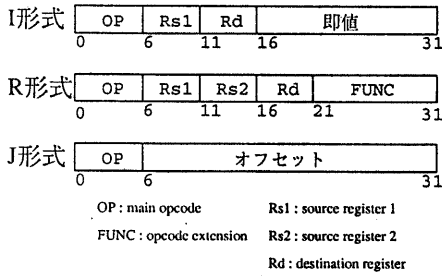


図 1: 命令フォーマット

下位 2 ビット	00	01	10	11
上位 4 ビット				
0000	R 型命令			
0001	J	JAL	TRAP	RFE
0010	LB	LBU	LH	LHU
0011	LW	LHI		
0100	SB	SH	SW	
0101	LF*	LD*		
0110	SF*	SD*		
0111				
1000	ADDI	ADDUI	SUBI	SUBUI
1001	ANDI	ORI	XORI	
1010	SLTI	SGTI	SLEI	SGEI
1011	SEI	SNEI		
1100	SLLI	SRLI	SRAI	
1101	BEQZ	BNEZ		
1110	BFPT*	BFPF*		
1111	JR	JALR		

\*:実装していない命令

図 2: 命令セット・オペコード (6 ビット)

下位	000	001	010	011	100	101	110	111
上位								
0000	ADD	ADDU	SUB	SUBU	AND	OR	XOR	
0001	SLT	SGT	SLE	SGE	SEQ	SNE		
0010	SLL	SRL	SRA		MULT	MULTU	DIV*	DIVU*
0011	ADDI*	SUBI*	MULTI*	DIVI*				
0100	ADDI*	SUBI*	MULTI*	DIVI*				
0101	CVTIF*	CVTI2D*						
0110	CVTIF2I*	CVTI2D2I*						
0111								
1000	LTI*	GTI*	LEI*	GEI*	EQI*	NEI*		
1001	LTI*	GTI*	LEI*	GEI*	EQI*	NEI*		
1010	MOVI*	MOVI*						
1011	MOVI2FP*	MOVI2FP*						
1100	MOVIAR2I†	MOVISR2I†	MOVBPI2I†	MOVBPD2I†				
1101	MOVI2IAR†	MOVI2SR†	MOVI2BPI†	MOVI2BPD†				
1110								
1111								

\*: 実装していない命令 †: 追加した命令

図 3: R 型命令におけるファンクションコード (7 ビット)

- 制御系レジスタ:
  - プログラム・カウンタ (PC)
  - ステータス・レジスタ (SR)
    - スーパーバイザ・フラグ, 旧スーパーバイザ・フラグ, 割込み許可フラグ, 旧割込み許可フラグ, 割込みベクトル (4 ビット) からなる。
  - 割込みアドレス・レジスタ (IAR): 割込まれた命令のアドレスを格納。
  - 命令/データ・アドレス・ブレイクポイント・レジスタ (BPI, BPD): アドレス・ブレイク機能 (追加仕様: 後述) のためのブレイクポイントを保持。
- 命令セット: データ転送, 算術論理演算, 分岐/ジャンプ, 浮動小数点演算の 4 種類に分類される。
- 命令形式: すべての命令は固定長かつ単一長, 命令形式は I 形式, J 形式, R 形式の 3 種類である。3 アドレス方式であり, 演算はレジスタ-レジスタ間およびレジスタ-即値間のみ可能である (ロード/ストア・アーキテクチャ), 図 1 参照。

- オペコード割当て: 割当ては我々が独自に与えている。オペコードは, OP フィールド (6 ビット) および (R 形式の場合) FUNC フィールド (11 ビット) で指定される。図 2 および図 3 に, オペコードの割当て結果を示す。なお, FUNC フィールドは 11 ビット中, 上位の 7 ビットしか用いていない。表中 "\*" 印は浮動小数点関係の命令であり, 実装していない。"†" 印は制御系レジスタ転送用として追加した命令を表わす。
- メモリ・アドレッシング: ビッグ・エンディアン方式, バイト, ハーフワード, ワード単位のアクセスが可能。アドレッシング・モードは, ベース相対および PC 相対 (ジャンプ命令のみ) が可能。

#### 4 命令パイプライン

文献 [6] に従い, 次の 5 ステージから成る命令パイプライン構成を採る。図 4 に QP-DLX のブロック図, および図 5 に, 各命令の命令パイプライン処理過程を示す。

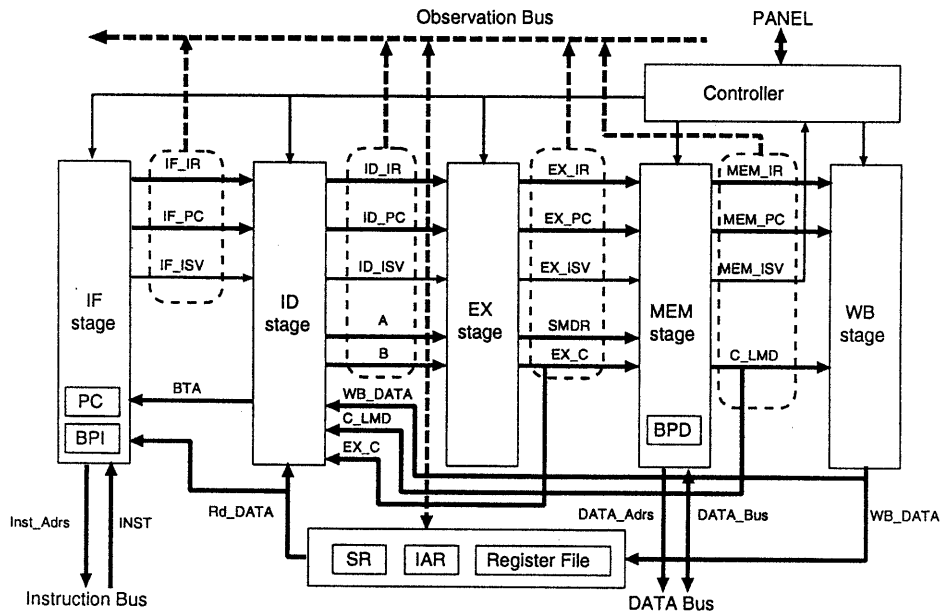


図 4: QP-DLX のブロック図

	IF	ID	EX	MEM	WB	
I 型 命 令	ロード命令	A←Rs1	C←A+((IR <sub>16</sub> ) <sup>16</sup> ##IR <sub>16,31</sub> )	LMDR←M[C]	Rd←LMDR	
	ストア命令	A←Rs1 B←Rd	SMDR←B	M[C]←SMDR	*****	
	ALU命令	A←Rs1	C←A op ((IR <sub>16</sub> ) <sup>16</sup> ##IR <sub>16,31</sub> )	*****	Rd←C	
	シフト命令	IR1←IR	C←A op ((IR <sub>16</sub> ) <sup>16</sup> ##IR <sub>16,31</sub> )	*****	Rd←C	
	セット命令	BTA←PC1+4+((IR <sub>16</sub> ) <sup>16</sup> ##IR <sub>16,31</sub> ) cond←(Rs1 op 0) if (cond) PC←BTA	*****	*****	*****	
	BEQZ命令	IR←Mem[PC]	PC←PC1+4+((IR <sub>16</sub> ) <sup>16</sup> ##IR <sub>16,31</sub> )	*****	*****	
	BNEZ命令	PC1←PC	A←Rs1	C←A	*****	R31←C
	JR命令	PC←PC+4	A←Rs1 B←Rs2	C←A op B	*****	Rd←C
R 型 命 令	ALU命令	A←Rs1	C←A	*****	Special←C	
	シフト命令	A←Special	C←A	*****	Rd←C	
	セット命令	PC←PC1+4+((IR <sub>6</sub> ) <sup>26</sup> ##IR <sub>6,31</sub> )	*****	*****	*****	
	MOV12S命令	A←Rs1	C←A	*****	R31←C	
J 型 命 令	MOV2I命令	A←Rs1	C←A	*****	R31←C	
	J命令	PC←PC1+4+((IR <sub>6</sub> ) <sup>26</sup> ##IR <sub>6,31</sub> )	*****	*****	*****	
	JAL命令	A←Rs1	C←A	*****	R31←C	
	TRAP命令	IVEC←int	*****	*****	IAR←PC1, PC←0	
RFE命令	RFE←1 PC←IAR	*****	*****	*****		

\*\*\*\*\*では何も行なわない

図 5: 命令パイプライン処理過程

1. IF：命令フェッチ
2. ID：命令デコード、レジスタ・フェッチ、および分岐先アドレス生成
3. EX：実行、および、実効アドレス生成
4. MEM：メモリ・アクセス
5. WB：書込み

各ステージは1クロックごとに動作を完了し、次のステージへデータを渡す。渡されるデータとして、命令(IR:32bit)、命令アドレス(PC:32bit)、割込みステータス・ベクトル(ISV:4bit)などがある。命令アドレスはPC相対番地計算のベース、および割込みアドレスを与える。割込みステータス・ベクトルは各ステージで発生する内部割込みイベントを記録する(5章参照)。各ステージからの出力されるデータ、およびレジスタ類の値を観測バス(Observation Bus)を通して外部から観測することができる(6章参照)。

DLXでは、RAW(Read After Write)データ・ハザード、制御ハザード、および、構造ハザードが生じ得る。ただし、QP-DLXではマルチサイクル命令を実装していないので、構造ハザードは生じ得ない。文献[6]に従い、RAWデータ・ハザードおよび制御ハザードに対して、以下のように対処する。

**フォワーディング**：後続3命令に対してフォワーディングを行なうため、図4に示すようにEX、MEM、WBの各ステージの出力EX.C、C\_LMD、WB\_DATAをIDステージに送る。ロード命令の場合、ロード遅延スロットが1個生じるが、当該ロード遅延スロットに位置する命令は、当該ロード結果をソース・オペランドとして使用できない。ロード遅延スロットより後の後続2命令に対して、ロード結果のフォワーディングが行われる。

**分岐**：文献[6]では分岐先アドレスBTAの生成をEXステージで行ない、PCの置換をMEMステージで行なう方法と、その改良版として、IDステージで分岐先アドレス生成とPC置換の両方を行なう方法が示してあるが、QP-DLXでは後者を採用している(図5)。そのため、分岐による遅延は1クロック・サイクルである。

ただし、遅延分岐を実現する場合、分岐遅延スロットで生じる割込みへの対処が複雑になるため、QP-DLXでは遅延分岐は実現しておらず、分岐遅延スロットの命令は分岐不成立時のみ実行される。

## 5 割込み

### 5.1 割込みイベント

QP-DLXでは以下の割込みイベントを実装する。表1に、割込みイベントの優先順位および割込みベクトルを示す。

#### ● 内部割込み：

- 命令ミスアラインメント：命令フェッチの際の命令アドレスの非整列化。
- 命令ページ・フォールト：命令フェッチの際のページ・フォールト。
- 命令メモリ保護違反：命令フェッチの際の不当な領域へのアクセス。
- 命令アドレス・ブレイクポイント：ブレイクポイントとして設定された命令アドレスからのフェッチ
- 未定義命令：定義/実装されていない命令の使用。
- 特権命令例外：ユーザ・モードでの特権命令の使用。
- トラップ命令：トラップ命令の使用。
- 算術割込み：加減算/乗算命令におけるオーバーフロー。
- データ・ミスアラインメント：データ・アクセスの際のデータ・アドレスの非整列化。
- データ・ページ・フォールト：データ・アクセスの際のページ・フォールト。
- データ・メモリ保護違反：データ・アクセスの際の不当な領域へのアクセス。
- データアドレス・ブレイクポイント：ブレイクポイントとして設定されたデータアドレスへのアクセス

優先順位	割込みベクトル	割込みイベント	外部/内部	同期/非同期	検出ユニット
1	1111	リセット	外部	非同期	割込みコントローラ
2	1110	命令ミスアラインメント	内部	同期	IFステージ
3	1101	命令ページフォールト	内部	同期	IFステージ
4	1100	命令メモリ保護違反	内部	同期	IFステージ
5	1011	命令アドレス・ブレイクポイント	内部	同期	IFステージ
6	1010	未定義命令	内部	同期	IDステージ
7	1001	特権命令例外	内部	同期	IDステージ
8	1000	トラップ命令	内部	同期	IDステージ
9	0111	算術割込み	内部	同期	EXステージ
10	0110	データ・ミスアラインメント	内部	同期	EXステージ
11	0101	データ・ページ・フォールト	内部	同期	MEMステージ
12	0100	データ・メモリ保護違反	内部	同期	MEMステージ
13	0011	データ・アドレス・ブレイクポイント	内部	同期	MEMステージ
14	0010	I/Oリクエスト	外部	非同期	割込みコントローラ
15	0001				
16	0000	割込みなし			

表1: 割込みイベント一覧

● 外部割込み：

- リセット：リセットキーが押された。
- I/O リクエスト：入出力装置からの要求。

デバッグ機能および観測機能を強化する目的でアドレス・ブレイク機能を追加している。命令アドレスとデータ・アドレスの双方について、あらかじめ指定されたアドレスをアクセスしたとき割込みを発生する。ブレイクポイントを記憶するために、命令アドレス・ブレイクポイント・レジスタ (BPI:32 ビット) とデータアドレス・ブレイクポイント・レジスタ (BPD:32 ビット) の2つを設けている。

5.1.1 割込み検出方式

QP-DLX では、「正確な割込み (precise interrupt)」, すなわち割込みを起した命令の先行命令は実行を完了しており、後続命令すべてはその実行を最初から再開できることを保証する。そして割込み処理を以下のような簡単な機構で実現している。

- 割込みイベント検出: 内部割込みイベントの場合、各イベントごとにそれが発生するステージにおいて発生の有無をチェックする。割込みイベントごとに発生するステージが定まっているため、各ステージにおいて割込みイベントの発生の有無をチェックする (表 1)。
- 優先順位判定: 命令パイプラインには命令 (IR) と共に割込みステータス・ベクトル (ISV) が流れており、各ステージで発生した内部割込みイベントのベクトルを ISV に書き込むようになっていいる。あるステージで内部割込みイベントが発生した場合、その上流ステージで優先順位の高い割込みイベントが生じ先に ISV にベクトルを書き込んでいないかを調べ、もし書き込まれていなければ当該割込みイベントのベクトルを書き込む。そしてその命令を無効化して状態更新を起させないようにする。WB ステージで発生する内部割込みはないため、MEM ステージの実行を終了して出ようとする命令の ISV を調べる。これにより最も先行する命令の内部割込みイベントが最初に認定されるようになる。
- 割込み処理: 毎クロック・サイクルごとに、MEM ステージの実行を終了した命令の ISV を調べ、その命令が内部割込みイベントを起しているか、あるいは外部割込みイベントがペンディングしていれば、次の割込み処理を行なう。
  1. パイプラインの全命令をフラッシュして無効化する。
  2. 内部割込みを起した命令、または外部割込みに割込まれた命令の命令アドレス (PC) を割込みアドレス・レジスタ (IAR) に書き込む。
  3. ステータス・レジスタ (SR) 内でスーパーバイザ・フラグおよび割込み許可フラグをそれぞれの旧フラグにコピーして退避を行なう。スーパーバイザ・モードおよび割込み禁止モードに入る。
  4. プログラム・カウンタにアドレス '0' を設定して、命令フェッチを行なう。
- 割込みからの復帰: RFE 命令によって、割込みアドレスレジスタ (IAR) のアドレスをプログラム・カウンタに設定し、スーパーバイザ・フラグおよび割込み許可フラグをそれぞれの旧フラグから復帰する。

6 観測機能

観測機能により、ソフトウェアの介入なしにプロセッサ内部の動作を観測する。観測機能は以下の3つからなる。

- レジスタの値の読み出し/書き込みを命令/データ用のバスとは独立に設けられた観測用のバス OB を通して行なう。
- 外部からの Start / Stop スイッチ信号によりプロセッサを動作/停止させる (動作/停止機能)。
- 停止状態から Step 信号により1サイクルの命令パイプライン実行後、停止状態に戻る (シングル・ステップ機能)。
- 割込み発生時に通常の割込み処理を行なう代わりに、停止状態に遷移する (割込み観測機能)。

全てのユーザ可視のレジスタ、および命令パイプラインの各ステージでラッチされている値を観測バスを通して読み出すことができる。観測対象の選択は O\_SEL (6 ビット) で行なう。表 2 に観測対象の一覧を示す。観測用バスを通しての書き込みは、汎用レジスタ、IF ステージのプログラム・カウンタ、およびステータス・レジスタ、ブレイクポイント・レジスタ、割込みマスク・レジスタに限られており、表 2 には \*印で示してある。

下位 2 ビット	00	01	10	11
上位 4 ビット				
0000	PC*	IF_IR	IF_ISV	
0001	ID_PC	ID_IR	ID_ISV	
0010	EX_PC	EX_IR	EX_ISV	
0011	MEM_PC	MEM_IR	MEM_ISV	
0100	WB_PC	WB_IR		
0101	SR*	IAR*	A	B
0110	SMDR	EX_C	C_LMD	Rd_DATA
0111	LSEL*	BPI*	BPD*	
1000	R0*	R1*	R2*	R3*
1001	R4*	R5*	R6*	R7*
1010	R8*	R9*	R10*	R11*
1011	R12*	R13*	R14*	R15*
1100	R16*	R17*	R18*	R19*
1101	R20*	R21*	R22*	R23*
1110	R24*	R25*	R26*	R27*
1111	R28*	R29*	R30*	R31*

表 2: 観測対象の指定 O\_SEL (\*は書き込み可)

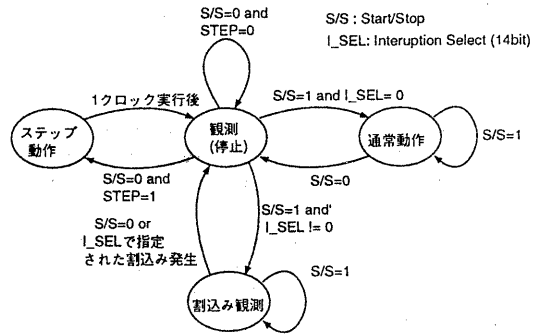


図 6: 観測機能の状態遷移図

動作/停止およびシングル・ステップ動作を行なうときの状態遷移図を図 6 に示す。これらを用いて内部のパイプライン動作の細かな観察、および非常に低速にプロセッサを動作させてパイプラインの流れを見せることを可能にしている。

割込み観測機能を用いて、割込み関係のソフトウェアのデバッグ、ブレイクポイントと組み合わせることによる通常のソフトウェア

ステージ	SFLの記述量(行) コメントを含む	ゲート数	備考
IF	686	1014	
ID	2483	18904 (16190)	内, レジスタファイル
EX	2247	19581 (15781)	内, 乗算器
MEM	490	977	
WB	351	394	

表 3: 各ステージの設計結果

のデバッグ, さらにキャッシュのミスヒットや仮想記憶のページ・フォールト等の現象の観測を行なうことができる。割込み観測機能において, 通常の割込み処理を行なうか, あるいは停止状態に遷移するかの選択は, レジスタ LSEL を通して行なわれる。LSEL には各割込みイベントに 1 対 1 に対応するフラグがあり, 1 つのフラグがセットされていれば, それに対応する割込みイベントが検出されたとき停止状態に遷移する。

## 7 開発状況

QP-DLX は, NTT で開発された論理合成システム PARTHENON 上のハードウェア記述言語 SFL を用いて開発されている。現在のところ, 通常動作部分の設計を終え, SFL 記述上での動作検証を行なっている。今後割込み制御および観測機能を実装する。SFL による設計が終了している各ステージの記述量, および論理合成結果を表 7 に示す。ここで記述量とは SFL によるソース・プログラムの行数であり, コメント行も含んでいる。ただし動作検証と回路修正により回路規模と記述量に多少の変動が有り得る。これまでに要している設計工数は, ミーティングやドキュメント作成のための時間を含めて, 228 時間・人である。

SFL による記述の完成後, 論理合成を行ない, 得られたネットリストに対し, 他の CAD システムでレイアウトを行なう。PARTHENON と結合可能な CAD システムとして, ES2 社の Solo1400 および COMPASS Design Automation 社のツールなどがあるが, 双方を用いてレイアウトを行ない, チップ面積等の結果の良いものを最終的に用いる予定である。製造は ES2 社に委託し, 1.0 $\mu$ m CMOS スタンドセル, 224 ピン PGA パッケージ (内, 電源を除く信号線 192 ピン) により, 20 個程度製造する予定である。

## 8 あとがき

以上, 現在進行中の「ソフトウェアから, 計算機アーキテクチャ, 論理回路設計, 集積回路設計・製造に至る計算機工学一貫教育」用マイクロプロセッサ QP-DLX 開発プロジェクトの方針, アーキテクチャ仕様, 開発状況について述べた。

今後, マイクロプロセッサおよびボードの製作, 設計データの整備・公開を順次行なってゆく予定である。本プロジェクトをより良いものとするため, 皆様からのご意見, ご指導が頂ければ幸いです。また, 集積回路設計・製造の教育に関しては, 大学だけの力ではその実現が非常に困難である。関係省庁, 企業等の皆様にぜひご協力頂きたいと心から願っている次第である。

## 謝辞

本研究の推進に当たっては, (株)物産システムテクノロジーおよび ES2 社, (株)ソリトンシステムズおよび Compass Design Automation 社, NTT データ通信 (株) および日本電信電話 (株) NTT

コミュニケーション科学研究所, 横河ヒューレットパッカード (株) および (株)YIIP システム技術研究所の各社に CAD ツールの提供などでご協力いただいております。この場を借りて御礼申し上げます。

日頃ご討論頂く九州大学 大学院総合理工学研究科 安浦研究室の諸氏に感謝致します。

なお, 本研究は一部, 文部省科学研究費補助金 試験研究 (B) 「計算機工学・集積回路工学教育研究用マイクロプロセッサの開発」(04555079) による。

## 参考文献

- [1] 江刺正喜, “大学での LSI 製作と教育,” 電子情報通信学会誌, vol.68, no.1, pp.50-52, 1985 年 1 月。
- [2] 神原, 安浦, “教育用マイクロコンピュータ KUE-CHIP の開発とその応用 — 集積回路技術を利用した情報工学実験 —,” 情報処理, vol.33, no.2, pp.118-127, 1992 年 2 月。
- [3] 庄野克房, “大学における LSI 設計教育 — 2 ビットマイクロコンピュータ —,” 電子情報通信学会誌, vol.75, no.5, pp.530-533, 1992 年 5 月。
- [4] 越智, 澤田, 岡田, 上嶋, 神原, 濱口, 安浦, “計算機工学・集積回路工学教育用マイクロプロセッサ KUE-CHIP2,” 信学技報, CPSY92-46, ICD92-86, 1992 年 10 月。
- [5] 末吉, 田中, 柴村, “再構成可能な論理 LSI を用いた教育用マイクロプロセッサ: KITE,” 信学技報, CPSY92-47, ICD92-87, 1992 年 10 月。
- [6] Hennessy, J. L. and Patterson, D. A., *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Inc., 1990;  
富田, 村上, 新實 (訳), ヘネシー & パターソン コンピューター・アーキテクチャー 設計・実現・評価の定量的アプローチ —, 日経 BP 社, 1992。
- [7] 諸富, 村上, 安浦, “計算機工学一貫教育用 DLX 風マイクロプロセッサの開発構想,” 電子情報通信学会技術研究報告, VLD92-83, 1993 年 1 月。
- [8] 諸富, “計算機工学一貫教育用マイクロプロセッサのプロトタイプ設計,” 九州大学工学部情報工科学卒業論文, 1993 年 3 月。