

教育用32ビットマイクロプロセッサ QP-DLX の ハードウェア記述言語による設計

1C-4

山家 陽 中村 秀一 岩井原 瑞穂 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

1 はじめに

我が国の大学における計算機工学教育は、その多くは講義のみによるもので、実験や実習を伴うものは極めて少ないのが現状である。しかも、ソフトウェアによる実現部分と集積回路技術によってハードウェアとして実現する部分を総合的に考える、システム設計側に立った集積回路設計という立場で組まれた実験カリキュラムはほとんど無いと思われる。そのため、ソフトウェア/ハードウェアの両面に渡る技術分野を概観できる一貫教育カリキュラムとして、システム設計を通じたソフトウェア/ハードウェア統合型実験/演習カリキュラムの開発が望まれる。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る「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 [2]

本稿では、現在開発中のマイクロプロセッサ QP-DLX (Kyushu University Education-Purpose DLX Microprocessor) について説明し、またハードウェア記述言語 SFL による設計について述べる。

2 QP-DLX プロジェクトの方針

[DLX アーキテクチャの採用] 単純、かつ、現在の技術水準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、文献 [1] の DLX に基づくことにした。

[教育のための機能] 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/停止を行なう機能からなる観測機能を備える。

[QP-DLX の設計データの公開] QP-DLX の設計ドキュメントおよび設計データを統合型 CAD ベンチマークとして広く公開し、論理合成や設計検証、レイアウトのベンチマークとして利用してもらおう。

3 QP-DLX のアーキテクチャ仕様

3.1 DLX アーキテクチャ

DLX は、文献 [1] 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。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

"HDL Design of the Education-Purpose 32 bit Microprocessor QP-DLX", Akira YAMAGA, Shuichi NAKAMURA, Mizuho IWAHARA, Kazuaki MURAKAMI, Hiroto YASUURA, Department of Information Systems, Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Kasuga-koen, Kasuga-shi, Fukuoka, 816 Japan; or e-mail to yamaga@is.kyushu-u.ac.jp.

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。以下、QP-DLX の仕様の概略を述べる。

- 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
- 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)
 - 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)
- 命令形式：全ての命令は固定長かつ単一長。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)。

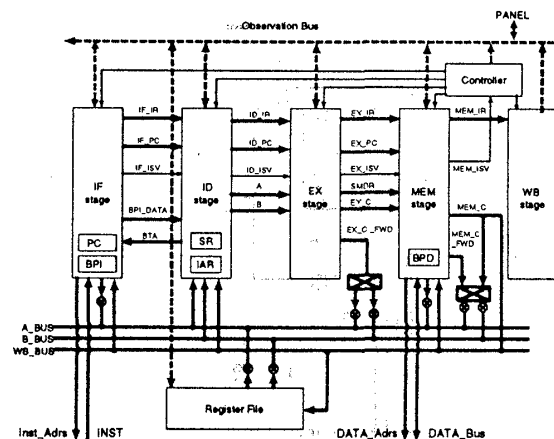


図 1: QP-DLX のブロック図

3.2 命令パイプライン

文献 [1] 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 1 に QP-DLX のブロック図を示す。なおコントローラは割込みおよび観測機能の制御を行なう。

- ①IF：命令フェッチ
- ②ID：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
- ③EX：実行、および、実効アドレス生成
- ④MEM：メモリ・アクセス
- ⑤WB：レジスタ書込み

3.3 観測機能

観測機能により、ソフトウェアの介入なしでのプロセッサ内部の動作の観測が可能になる。観測機能は以下の4つからなる。

- 全てのユーザ可視のレジスタ、命令パイプラインの各ステージでラッチされている値、および信号線を命令/データ用のバスとは独立に設けられた観測用のバスを通して読み出すことができる。全てのレジスタには書き込みも可能である。
- 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)。

4 開発過程

図2にQP-DLXの設計フローを示す。

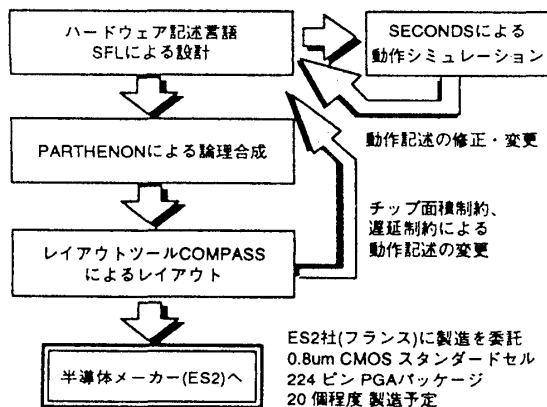


図2: 開発過程

5 ハードウェア記述言語 SFL による設計

QP-DLXの設計は、NTTで開発された論理合成ツール PARTHENON 上のハードウェア記述言語 SFL を用いて行なった。QP-DLX の SFL による設計およびドキュメント作成は3人で行ない計688時間・人を費やした。各モジュールの SFL による記述量は後続の論文にゲート数およびレイアウト面積と共に記述した。

5.1 SFL による状態遷移機械の設計

QP-DLX は図3に示す状態遷移図に従って動作する。QP-DLX は2つの状態遷移で構成され、上位の状態遷移は観測を行なうための動作に対応し、下位の状態遷移はメモリアクセス時の動作に対応する。上位の状態遷移と下位の状態遷移は直積の関係にある。上位の状態遷移を除くと観測機能のない通常のプロセッサの動作に相当する。このような状態遷移機械をハードウェア記述言語で設計するには、①並列に動作する2つの状態遷移機械が記述できなければならない、かつ②状態遷移機械同士が互いに相手と通信または制御を行な

うことができなければならない。SFL では①に対しては、並列動作を記述する stage 文および task 文を用いて記述することができ、またそれぞれの状態遷移機械における状態遷移も状態を記述する state 文および状態遷移を記述する goto 文を用いて記述することができる。さらに②に対しても、制御内部端子文を状態遷移機械間の通信を行なうために用いることにより記述できる。以上のように SFL は状態遷移機械を記述するのに適した言語であり、QP-DLX の状態遷移機械を容易に記述することができた。

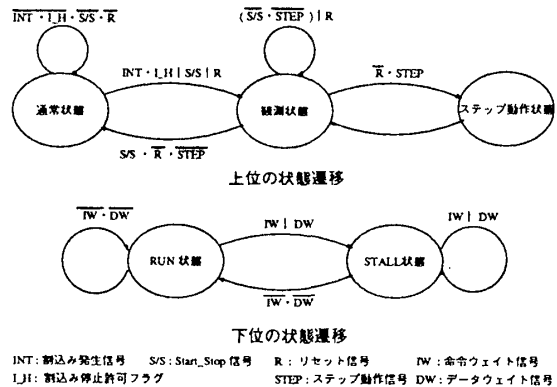


図3: QP-DLX の状態遷移図

5.2 SFL の問題点

SFL でのデータバス部の記述がソースを見ただけでは実際にどのようなセクタあるいはバスとして合成されるかわかりづらい。またゲート数だけではレイアウト面積の見積もりが困難であったため合成時にデータバスの量を評価し報告する機能が望まれる。また、これはテキスト型言語一般にいえることであろうが SFL ではサブモジュール間で端子同士を接続する場合に制御信号の引数渡しで接続することが多い。このため、SFL ソースを一見しただけではどの端子とどの端子が接続されているかわかりづらい。この解決のためモジュール間接続の図形表示を行なう視覚的なインターフェイスが望まれる。

6 おわりに

以上、教育用マイクロプロセッサ QP-DLX のプロジェクト、アーキテクチャおよびハードウェア記述言語 SFL による設計の利点および問題点について述べた。

参考文献

[1] 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・アーキテクチャ 設計・実現・評価の定量的アプローチ, 日経 BP 社, 1992.
 [2] 岩井原他, “教育用計算機 QP-DLX の開発と開発環境,” 電子情報通信学会技術研究報告, VLD93-86, 1993年12月。