

Hardware/Software Co-design 実験による システムアーキテクチャ設計学習プログラム

三井 浩康[†] 清尾 克彦^{††} 遠藤 聡[†] 神戸 英利^{†††} 小泉 寿男[†]
[†]東京電機大学 ^{††}株式会社ゼネテック ^{†††}三菱電機株式会社

組込みシステム製品の多機能化、小型化・省電力化、開発期間短縮が急速に進んでいる。新たな開発手法が模索されているが、多機能化による組込みシステムのソフトウェア技術者の不足が重大な問題となっている。とくに、高い技術をもつシステムアーキテクトやソフトウェア技術者の育成への関心が強く、大学での組込みシステム教育への期待が高い。本論文では、ハードウェア技術のわかる組込みソフトウェア技術者の教育を目指して、新しい開発手法であるハードウェア・ソフトウェア協調設計を用いた組込みシステムアーキテクチャ設計の学習方式を提案し、その評価について述べる。

Learning Program of Embedded System Architecture Design Based on Hardware/Software Co-design Experiment

HIROYASU MITSUI[†] KATSUHIKO SEO^{††} SATOSHI ENDO[†] HIDETOSHI KAMBE^{†††} HISAO KOIZUMI[†]
[†]Tokyo Denki University ^{††}Genetec Corporation ^{†††}Mitsubishi Electric Corporation

Embedded systems have been evolving to meet the requirements for functional diversity, downsizing, low power consumption, and short life cycle. Although new development methods have been researched, functional diversity has caused an important issue that the number of embedded software engineers is not enough. Especially, training system architects and software engineers with high ability becomes a big concern and enterprises expect embedded system education in universities. In this paper, aiming to train embedded software engineers who understand hardware, we propose a learning method of embedded system architecture design through an experiment based-on a new development method of hardware/software co-design, and describe its evaluation.

1. はじめに

組込みシステム (Embedded System) は、携帯電話、情報家電、AV 家電、自動車、ロボット、産業機器などさまざまな分野に適用されている。企業間競争の激化が製品の多機能化・高性能化を助長し製品の改版を早める中で、企業は機能・性能の向上、開発期間短縮、コスト削減のための新たな開発手法を模索している。小型化・省電力化のためのハードウェア(以下、HW と略す)の重要性とともに、多機能を実現するための組込みソフトウェア(以下、SW と略す)の重要性が非常に高まっており、HW と SW を統合開発して所要期間とコストを削減する Hardware/ Software Co-design (以後、HW/SW 協調設計と略す)技術および SW の設計品質向上技術が注目されている。

一方、組込み SW の重要性にもかかわらず、経済産業省の調査によれば、2006 年度の我が国の組込み SW 技術者数は約 23.5 万人であり、約 9.9 万人の不足といわれている[1]。この傾向はさらに顕著になる傾向にある。なかでも技術レベルの高い技術者、HW を理解できる SW 技術者が不足しており、その育成が組込みシ

ステム産業における大きな課題となっている。

組込み SW 技術者教育は、これまでは企業内で行われる場合が多かったが、近年は大学教育への期待が強い。経済産業省の調査によれば、理工系大学では組込みシステム教育への関心はあるが、体系立った教育コースを開設している大学は少なく、情報工学系学科、電気・電子系学科で実施している例も多い[2]。

大学での組込みシステム技術教育は、従来はシステム LSI 設計技術あるいは CPU アーキテクチャ技術など HW 寄りの要素が強かった。組込み SW 技術者教育では、情報工学系科目を中心とする SW 開発技術のほかに HW 基礎知識、HW/SW Co-design を含む基礎教育が必要である。将来、組込み SW 技術者あるいはアーキテクトになる可能性のある学生を HW 技術のわかる組込み SW 技術者に育成することが、大学における組込み SW 技術教育の課題といえる。

HW/SW Co-design は、HW、SW を区別しない抽象度の高い機能設計から始めて、HW 開発と SW 開発を統一された環境下で並行的に、かつ相互に協調させながら進める新しい開発設計手法であり、システム記述言語として SpecC や SystemC が使用される。

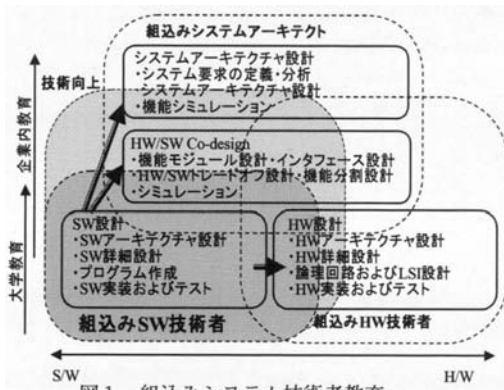


図1 組み込みシステム技術者教育

組み込みシステム技術者は抽象度の高いシステム設計をおこなうシステムアーキテクトとシステム設計を実際のHW, SWに具体化するHW技術者, SW技術者に分類される。図1はこれらの技術者の教育の層別である。SW技術者の技術レベルは上級・中級・初級技術者に分類される[3]。組み込みSW技術者はシステムアーキテクトの設計を理解して、SWとHWのインタフェース設計、対象HWの制御をおこなう必要がある。このため、システムアーキテクト設計手法、HW/SW協調設計およびHWに関する技術・知識の習得が必須となる。大学では初級技術者の教育が主となる。

本稿では、「HW技術のわかる組み込みSW技術者」の育成を目標に、システムアーキテクト設計およびHW設計技術の基礎を習得させるHW/SW Co-design実験方式とそれを用いた学習プログラムを提案する。さらに実証をおこなった結果について述べる。

2. 組み込みソフトウェア技術教育の現状と課題

2.1 組み込みシステム製品の発展とSWの役割増大

組み込みシステムのCPUには4ビットから32ビットさらにはマルチチップまで多様であるが、主力は8ビットまたは16ビットの単一CPUである。組み込み用のリアルタイムOS(以下、RTOSと略す)にはITRON(the real-time operating system for inclusion), 組み込みLinuxなど数十種類がある。画像処理などの高速処理にはDSP(digital signal processor)やASIC(application specific integrated circuit), FPGA(field programmable gate array)などが使用されている。

近年の組み込みシステムでは、複雑な機能のものやLAN, 無線でネットワーク化するものが増えており、多機能化のために組み込みSWの役割が増大している。SWを開発する言語はC, C++, Visual C++, Visual Basicなどが主である。このほかにシステム仕様や抽

表1 組み込みSW技術者に必要な技術基盤

	SW技術	HW技術	システム技術
従来の情報工学教育	<ul style="list-style-type: none"> ・プログラム言語 ・OSドライバ ・データベース ・ソフトウェア工学 ・通信プロトコル ・アドホック通信 	<ul style="list-style-type: none"> ・論理回路 ・コンピュータアーキテクチャ ・言語とオートマトン 	<ul style="list-style-type: none"> ・ネットワーク(LAN, WAN) ・クライアント/サーバシステム
	<ul style="list-style-type: none"> ・プログラム演習, TCP/IP, データベース ・情報工学実験(アーキテクチャ, FPGA) 		・情報ネットワークシステム
組み込みSW教育	<ul style="list-style-type: none"> ・リアルタイムOS ・組み込みSW設計 ・品質向上技術 	<ul style="list-style-type: none"> ・LSI(FPGA)設計 ・センサ ・無線/遠隔制御 	<ul style="list-style-type: none"> ・システム設計(要求分析・モデリング) ・HW/SW Co-design ・アーキテクチャ設計 ・SystemCによる設計/シミュレーション ・HW/SWトレードオフ
	<ul style="list-style-type: none"> ・HW/SW協調設計 ・分散型組み込みシステムネットワーク 		

象度の高い機能モジュールを記述するSpecCやSystemCなどのシステム記述言語がある。

2.2 大学における組み込みシステム教育の課題

組み込みシステム技術の教育では、これまでプロセッサおよびシステムLSIの設計技術教育の比重が高く、学生実験でも教育用プロセッサ設計やLSI設計など、HW分野の技術習得が主であった。情報工学では、情報数学、コンピュータ科学、基本ソフトウェア、コンピュータネットワーク、情報処理、ソフトウェア工学、マルチメディア工学などの講義およびプログラミング演習をおこなう。実験では、コンピュータアーキテクト、IC回路、FPGA回路などの実験のほか、ライントレースロボット構築実験などの例があるが、組み込みSW技術者向けに体系化された事例はまだ少ない。

表1は筆者らの研究[4][7]、情報処理推進機構(IPA)がまとめた「組み込みソフトウェア開発のための組み込みスキル標準ガイドブック」[5]および他大学での事例[3][6]を参考に、HW技術のわかる組み込みSW技術者育成の技術要素として整理したものである。従来の情報工学系科目に組み込みSW技術系科目を加えたこれらの基礎技術をいかに教育するかが課題である。

3. 学部における組み込みSW技術教育手法の提案

3.1 提案のねらい

本研究の目的は、システムアーキテクト設計、HWとSWのインタフェース(以下、I/Fと略す)設計、HWとSWのトレードオフ設計のわかる組み込みSW技術者を育成する技術習得方式を提案することである。

本提案では、表1の組み込みシステムの技術要素を段階的に習得させるために、図2に示すように組み込みSW開発、組み込みHW開発、HW/SW Co-design(協調設計)の各実験を段階的に実施する。本稿では、この中で、HW-SW Co-design実験を用いたシステムアーキテクト設計の学習プログラムについて述べる。

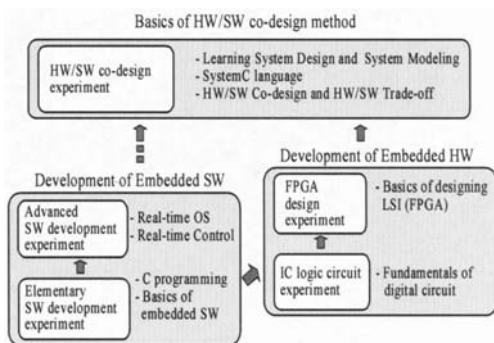


図2 組み込みシステム技術習得実験

3.2 SystemCによるHW/SW Co-design 学習

3.2.1 SystemCの概要

H/W-S/W Co-designは統合開発環境下でHWとSWの設計を並行しておこなう。抽象レベルの高いシステムアーキテクチャをHWおよびSWに具体化する際に設計初期段階からHWとSW間の協調の開発・検証を実施でき、品質向上と開発期間の短縮が図れる[7]。

SystemCはC言語を拡張したものであり、SWモジュールを記述できるだけでなく、HWモジュールを記述するための並列動作や同期、通信、タイミングなどの構文を含む[8]。SystemCでは異なる抽象度のモジュール記述が可能であり、SWとHWの区別のない高い抽象レベルから始めて抽象度を変えながら設計を進めることができる。また、モジュールの機能部とインタフェース部の設計を分離できる。

3.2.2 SystemCによるHW/SW Co-designの流れ

SystemCによるHW/SW Co-designの流れを図3に示す。組み込みシステムは組み込みCPUと専用H/W、およびRTOS上で動く複数のタスクで構成される。HW-SW Co-designではこれらを統合して、システムモデリングから詳細モジュールの設計までをおこなう。はじめにシステム要求仕様書からUTFモデル、TFモデルを用いてHW、SWの概念のないTLM(Transaction Level Modeling)としてシステムをモデリングする。TLMではSystemCのI/Fクラスを使って、抽象レベルの高いチャンネルでモジュールを接続する。

(1) UTFモデル:システムを機能面から複数モジュールに分割し、最も抽象度の高いUnTimed Functional (UTF)モデルとしてSystemCで記述する。モジュールは時間的要素のない機能モデルとして記述され、モジュール間はFIFO(First In First Out)チャンネルで接続される。UTFモデルでは、入出力、アルゴリズム処理の時間をゼロとして処理する。この段階でのシミュレーションでは設計したモデルが機能仕様を満たすことを確認する。

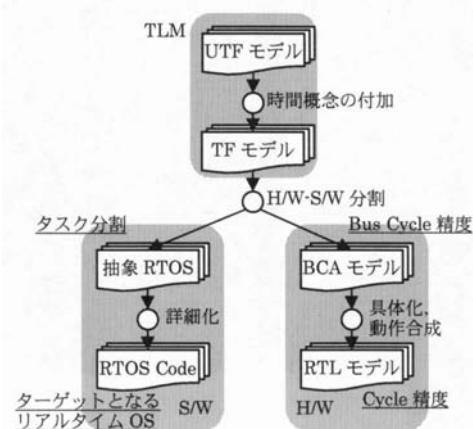


図3 SystemCによるHW/SW Co-designの流れ

(2) TFモデル:つぎに、UTFモデルに時間的要素を付加し、時間経過に従って入出力やアルゴリズム処理を進めるためのTF(Timed Functional)モデルを記述する。この段階では各モジュールの性能をシミュレーションする。

(3) HW/SWの分割:TFモデルの性能シミュレーション結果に基づいてモジュールをHWまたはSWに分割する。モデルの改良を繰り返し、最適なシステム構成を設計する。

(4) H/W設計/BCAモデル:HW化するモジュールの抽象度をBCA(Bus Cycle Accurate)段階に下げる。この段階では、通信部のみをクロックを用いたBCAモデルとして記述する。

(5) H/W設計/RTLモデル:BCAモデルから動作合成するか、または設計者がRTL設計をするか、いずれかの方法でモジュールの機能部をRTL(Register Transfer Model)モデルとして生成する。RTLモデルを論理合成してゲートレベルに変換してネットリストを生成し、LSIを製造する。

(6) S/W設計:S/Wモジュールは、RTOS上のタスクとして実現される。

3.3 HW/SW Co-design 実験の流れ

本提案では、HW/SW Co-designの教材として、静止画像処理で使用されるJPEGエンコーダを採用する。JPEGエンコーダのHW/SW Co-designは初めて実験する学生には難解であるので、実験を第一段階と第二段階に分けて段階的に実施する方式とした。第一段階では簡単な事例でHW/SW Co-designの流れ、概要を理解する。第二段階では複数のモジュールに分割されるJPEGエンコーダを対象としてHW/SWトレードオフ設計を含むHW/SW Co-designを体験し理解を深めることにした。実験の環境を表2に示す。

表 2 HW/SW Co-design 実験の環境

開発プラットフォーム	Windows XP
プログラミングツール	Microsoft Visual C++
システム記述言語	SystemC 2.0
シミュレーションツール	GTKWave
デバイス構成ツール	QuartusII Web Edition
デバイス SW 開発ツール	NiosII IDE
デバイス	NiosII CycloneII

4. HW/SW Co-design 学習プログラムの実証

4.1 実験の手順

提案する HW/SW Co-design 実験の手順を図 4 に示す。学習プログラムの実証を以下の手順で進めた。

(1) 要求仕様書から TLM をつくる：はじめにシステム要求仕様書からモデリングをおこなう手法を学ぶ。UTF モデル、TF モデルを理解させ、HW、SW の概念のない TLM としてシステムをモデリングする。この段階では、抽象モジュールを HW、SW モジュールに自動変換するツールを採用する。このため、TLM でモデリングするための規約に従って、学生に SystemC で UTF/TF モデルを記述させる。

記述したモデルのシミュレーションをおこない、入力に対して正しい出力がされるかを確認させる。

(2) H/W-S/W トレードオフ解析とモジュール分割：ここでは、H/W-S/W トレードオフ解析をおこない BCA レベルでの HW 設計を経験させる。

(3) 動作合成、論理合成：BCA レベル設計を元に RTL レベル設計に変換する。

(4) デバイス上で動作する S/W の開発：NiosII IDE SW 開発ツールを用いて Nios 開発評価ボード上に実装する SW を記述し、コンパイルおよびボードへの書き込みをおこなう。

(5) 実機検証：開発したシステムの実装は Nios 開発評価ボード上でおこなう。

4.2 第一段階 (RGBtoYUV 変換システムの設計)

JPEG エンコーダの色変換部 (RGBtoYUV 変換システム) を教材にして HW/SW Co-design 基礎実験をおこなった。RGBtoYUV 変換システムは静止画像の RGB データを演算によって YUV データに変換する。RGB データは R(赤)成分、G(緑)成分、B(青)成分それぞれを 8 ビットで表現する。YUV データは Y(輝度成分)、U(色差成分：青成分-輝度成分)、V(色差成分：赤成分-輝度成分)をそれぞれ 8 ビットで表現する。

RGB から YUV への変換はつぎの式でおこなう。

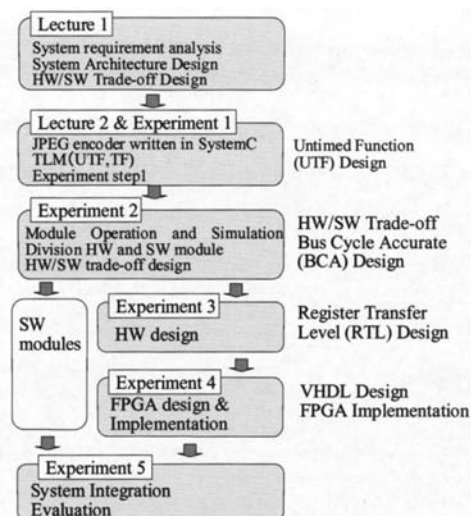


図 4 HW/SW Co-design 実験の手順

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

$$U = 0.564 \times (B - Y) + 128$$

$$V = 0.713 \times (R - Y) + 128$$

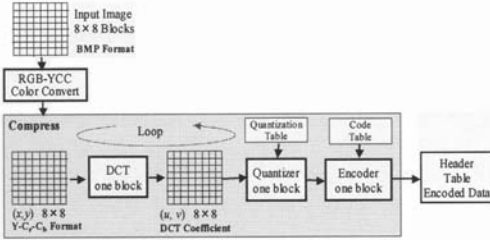
この色変換システムを HW/SW Co-design の流れに沿って設計し基本的な方法を学ぶ。HW/SW トレードオフ解析については、RGBtoYUV 変換システムを HW 化するとして HW 設計を経験させた。

4.3 第二段階 (JPEG エンコーダシステムの設計)

複数のモジュールで構成されるシステムとして JPEG エンコーダを設計対象に HW/SW Co-design を体験させた。この段階ではモジュール性能の解析に基づく HW/SW トレードオフ設計についても学習する。

4.3.1 JPEG エンコーダの概要

JPEG エンコーダは、図 5 に示すように色変換、DCT (discrete cosine transform, 離散コサイン変換)、量子化、符号化の各モジュールで構成される。色変換は第一段階で述べた RGBtoYUV 変換である。DCT 処理は、静止画像を 2 次元の周波数成分の組み合わせで表わし、画像の圧縮をおこなう。静止画は低周波成分が主であるので高周波成分を除去することでデータ量を大幅に圧縮できる。DCT 処理部は図 5 に示す演算式を用いて演算するが、繰り返し演算を実行するため、もっとも処理時間がかかる。本実験では DCT を HW/SW トレードオフ設計の対象として設定する。使用する JPEG プログラムは IJG (Independent JPEG Group) が配布しているソースプログラム [9] から必要最小限の機能を抽出して SystemC で記述した。



$$F(u,v) = \frac{1}{8} C(u)C(v) \sum_{x=0}^7 f(x,y) \cos \frac{\pi x u}{2 \times 8} \cos \frac{\pi y v}{2 \times 8}$$

$$= \frac{1}{8} C(u) \sum_{y=0}^7 \left(\frac{1}{8} C(v) \sum_{x=0}^7 f(x,y) \cos \frac{\pi x (2x+1)}{2 \times 8} \right) \cos \frac{\pi y (2y+1)}{2 \times 8}$$

$$c(n) = \begin{cases} \frac{1}{\sqrt{2}} & (n=0) \\ 1 & (n \neq 0) \end{cases}$$

図5 JPEG エンコーダの処理概要

4.3.2 実験の手順

実験は、図4に従って講義2回、実験5回の7回でおこなった。各回はいずれも2コマ(4時間)とした。第1回目の講義では、システム要求分析手法、システムアーキテクチャ設計手法および機能モジュール設計とHW/SW機能分割手法を解説した。またJPEGエンコーダの概要、色変換処理、DCT処理の概要を解説した。第2回目は、色変換システム、HW/SW協調設計の流れを解説しながら第一段階の実験を実施した。第3回目以降は、第二段階の実験として、JPEGエンコーダのHW/SW Co-designを実験を通して学習した。実験では、SystemCで記述したJPEGエンコーダのSW機能モジュール群をあらかじめ用意して与えた。

トレードオフ解析をおこない、DCT機能モジュールをHW化してFPGAに実装し、評価をおこなった。

4.3.3 トレードオフ解析とHWモジュールの設計

トレードオフ解析では、各機能をH/WまたはS/Wのどちらで実現するか、評価指標を設定し解析する。一般的にはLSIなどのH/WはS/Wに比べ性能面では勝るが、コスト面では劣る。小型・省電力化、性能、コストなどのさまざまな指標でトレードオフ解析をする手法を講義で解説した。実験ではモジュールの性能を指標にトレードオフ解析を学習することとし、HW化の対象モジュールはDCTとした。HW/SWモジュール自動変換を利用してDCTモジュールをH/Wに変換し、順次、抽象度を下げてHWの設計を進めた。

図7にDCTモジュールのHW設計の過程で、UTFレベルからBCA、RTLレベルに抽象度を下げて設計していく様子を示す。DCT処理にはloefflerのアルゴリズムを使用した[10]。

S/Wモジュールは、RTOSのタスクとして実現し、システムをNiosII開発評価ボード上に実装した。

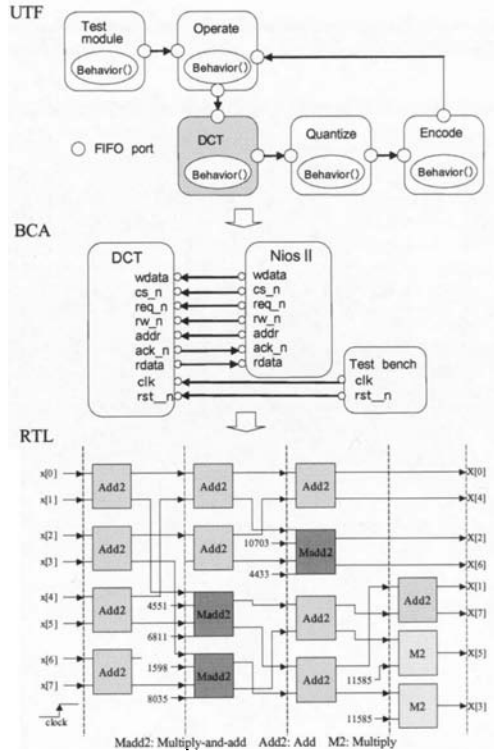


図6 DCTモジュールのHW設計

5. 評価と考察

筆者らの研究室の学生7名を被験者に実証実験をおこない、終了後にアンケート調査とレポート作成、理解度テストを実施した。アンケート調査結果の一部を表3に示す。また、レポートとテストの結果を表4に示す。アンケート調査では、5段階評価でSystemCによるシステム設計の講義の理解度が3.2、HW/SW協調設計の理解度が3.8であり、講義の理解度がやや低かった。レポートの平均得点は67.5点、小テストの平均得点は68.8点であり、学生は理解を深めたが、60~70点の成績が多く、実験内容が難しかったといえる。これはアンケートによる難易度の評価が4.2と高かったことでもわかる。

本実験では、C言語の知識だけでなく、HWの構造や記述法に関する知識も必要であり、性能測定ではCPUの原理や動作の知識も必要となる。とくに、HWの記述はC言語による逐次処理に慣れていないと理解が難しいことがわかった。HW/SW協調設計では、抽象度の高いレベルから設計をおこなえるが、最終的にはHWの知識が必要となる。これらが難易度を高めた要

表 3 HW/SW Co-design 実験のアンケート調査結果

Evaluation Items	Evaluation distribution					Mean
	5	4	3	2	1	
1. What was your understanding about the lectures for system design?	0	1	4	0	0	3.2
2. What was your understanding about the flow of HW/SW co-design?	1	2	2	0	0	3.8
3. What was your understanding about HW/SW trade-off design?	1	2	2	0	0	3.8
4. Was the HW/SW co-design experiment	2	2	1	0	0	4.2
5. Was the experiment interesting?	1	3	1	0	0	4.0
6. Was the lecture preceding the experiment useful?	2	2	1	0	0	3.8
7. Will the experiment be useful for embedded system education?	3	3	0	0	0	4.4

5: excellent, 4: good, 3 acceptable, 2: marginal, 1: poor

表 4 HW/SW Co-design 実験の理解度

Evaluation Items	Evaluation distribution					Mean
	90~	80~	70~	60~	~60	
Report score	0	1	1	4	0	67.5
Test score	0	1	2	2	1	68.8

因である。当初は、JPEG エンコーダの設計をいきなり始めたが、難度の高い実験となった。このため、準備段階の実験として第一段階の色変換部の設計を追加した。実業務における HW/SW Co-design は大規模で実際に体験することは困難であり、システムアーキテクチャ設計の学習として JPEG エンコーダの HW/SW Co-design を体験することは有意義と考える。

RGBtoYUV変換の機能モジュールは2つで、1クロックで1サイクルの動作が終了する。被験者からは、設計が容易であった、SystemCやVHDLの記述方法を学ぶ上で有効であった、HWとSWの通信、割込みについて理解したなどの意見が得られた。

第二段階の実験では、乗算が多く複数のクロックで1サイクルの処理をおこなうので、被験者は並列処理の同期に苦労した。トレードオフ解析については、システムをHWとSWに分割する考え方と手法を理解したとの意見が得られたが、性能以外の評価指標についての具体例による説明の追加が必要と思われる。アンケートで有効との評価が3.8と高かった事前講義を強化して、実験に必要なHWの基礎知識を向上するとともに、実験テキストの充実、実験内容の見直しにより実験の難易度を下げ、学生の理解度を向上させる改善が必要である。

提案実験は、初級組み込みSW技術者に組み込みシステムアーキテクチャ設計を経験させるのに有効であったと考える。従来の情報工学系実験でSW開発実習、組み込みSW開発実験、FPGA論理設計実験などが実施されてきたが、HWのリアルタイム制御実験、HW/SW Co-design 実験などの組み込みシステムに特有の技術習得実験および講義を加えることで従来の情報工学系の

実験および講義に比べて組み込みシステム技術の基礎習得範囲が拡大したと考える。

6. まとめ

組み込みシステム開発で重要性を増しているHWのわかる組み込みSW技術者育成を大学教育として実施することを目標に、組み込みシステムのシステムアーキテクチャ設計、HW/SW Co-design の基礎を習得する学習方法を提案した。JPEG エンコーダを使用した HW/SW Co-design 実験を提案し、システムを構築して実証した。今後の研究課題としては、以下の事項がある。

- ・HW/SW 協調設計実験の学生の理解度向上のために実験内容・実験テキストの見直し、事前講義の強化、他の講義科目との連携強化を図る。
- ・提案した HW/SW 協調設計学習プログラムを実際の授業に適用して評価する。

参考文献

- [1] 経済産業省商務情報政策局（監修：組み込みソフトウェア開発力強化推進委員会）：「2007年版組み込みソフトウェア産業実態調査報告書第1版改訂、-経営者・事業責任者向け調査」、2007
- [2] 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター：「2005年版組み込み関連技術教育実態調査報告書」、2005
- [3] 山本雅基，阿草清滋，間瀬健二，高田広章，河口信夫，富山宏之，本田晋也，金子伸幸：大学における社会人向け組み込みソフトウェア技術者人材養成の実施と分析，SEC Journal, No. 4, pp. 36-45, 2005
- [4] Hiroyasu Mitsui, Hidetoshi Kambe, Hisao Koizumi: et.al: 'A Student Experiment Method for Embedded System Education Based on Incremental Upgrade', Proc. of ADPNA2007 Workshop, ICPP2007.
- [5] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター監修：「組み込みソフトウェア開発のための ETSS 標準ガイドブック」,日経BP社, 2006
- [6] 有田五次郎：「組み込みソフトウェア技術者育成実践教育プログラム」実施報告，産学協同実践的IT教育支援事業，九州産業大学情報学会誌，4巻，1号，pp. 2-10, 2005
- [7] 遠藤祐，小泉寿男：「再利用モジュールのオンライン評価を取り入れたハードウェア・ソフトウェア協調設計方式とその検証」,電気学会論文誌C部門，Vol. 124, No. 11, pp. 2249-2259, 2004
- [8] D. C. Black, J. Donovan, 中西早苗訳：「基礎から学ぶ SystemC (SystemC : From The Ground Up)」, 丸善株式会社, 2006
- [9] Independent JPEG Group, <http://www.ijg.org/>
- [10] C. Loeffler, A. Ligtenberg and G. Moschytz: "Practical Fast I, D DCT Algorithms with 11 Multiplications", Proc. ICASSP '89, pp. 988-99, 1989