

VGAによる動作可視化機構を備えたプロセッサ教育支援システム

An Educational Microprocessor System with Visualization Mechanism using VGA

山脇 彰†, 芹川 聖一†

Akira Yamawaki and Seiichi Serikawa

1 はじめに

プロセッサはあらゆる電子機器に搭載されており、工学教育上、その理解は重要なことから、多くのプロセッサ教育支援システムが開発されている [1, 2, 3, 4, 5, 6].

シミュレータを用いた教育支援システム [1, 2] は、パーソナルコンピュータ (PC) があれば利用可能であり、汎用性に富んでいる。ただし、プロセッサの動作はソフトウェア上の仮想空間のできごとであり、コンピュータの動作原理に理解がない初心者にとっては、画面を通して何を観測しているのか直感的にわかりにくい。

一方、FPGA とそれを搭載したボードを用いた教育支援システム [3, 4, 5, 6] では、ハードウェアに触れながら自らプロセッサを動かすことによってノイマン型プロセッサの動作原理を直感的に理解しうる。さらに、ハードウェア記述言語 (HDL) によるプロセッサの実装を通して、プロセッサを構成するデジタル回路やその設計法の理解にも役立つ。しかしながら、専用ボードの入手が困難であったり、表示能力がボード上の LED や 7セグメント LED のみで乏しかったりする。

そのような背景から、様々な市販 FPGA ボードに実装でき、かつ、プロセッサ内部の表示能力に富んだプロセッサ教育支援システムを開発する。表示に関しては、多くの市販 FPGA ボードが VGA ポートを持つため [7, 8, 9, 10, 11]、ディスプレイを用いた可視化機構を設ける。ディスプレイ上にはレジスタの内容だけではなく、データパス上の内容も表示させ、クロックに同期しながらレジスタ間をデータが如何に移動するのをも可視化する。また、教材として設計データが容易に入手でき、教育目的に応じて自由に変更できることが望ましいことから、HDL プログラムを Web 上で公開し、自由に入手かつ変更できるようにする (<http://www.ds.ecs.kyutech.ac.jp/>)。

以降、2章において、開発したプロセッサ教育支援システムの概要について述べる。そして、3章において、VGA による動作可視化機構の詳細を示す。4章の実験では、いくつかの FPGA ボードへの実装を通してプロセッサ教育支援システムの汎用性を評価する。最後に5章でむすぶ。

2 プロセッサ教育支援システム

2.1 教育課題と本システムの目的

プロセッサは、一般的にレジスタトランスファレベル (RTL) で設計される。RTL では、レジスタ間でのデー

タ転送を主体にデジタル回路が設計される。つまり、プロセッサの動作原理を理解させるには、単にレジスタの値だけではなく、レジスタ間のデータパスを可視化することが重要である。それには、比較的大きな可視化装置が必要であり、従来からのボードシステムでは前節で述べたように限界があった。そこで、本システムでは表示領域の大きな VGA ディスプレイ上にプロセッサのデータパスまでも表示させる。当然、プロセッサの動作が目視できるように、スイッチによってステップクロックを入力できるようにする。ただし、ボードクロックとステップクロックをプロセッサのプログラム上で切り替えられるようにし、必要な個所のみ細かな動作を目視できるようにする。以上のことから、プロセッサの動作や RTL でのハードウェア設計の原理をより深く理解させることを目指す。

教材としては、前節で述べたように、講義や自習などを目的に自由に入手でき、かつ、簡単に使用できることが望ましい。本システムに関しては、設計データである HDL プログラムだけではなく、コンパイル済みの FPGA 構成データ、および、アセンブラや PC と FPGA ボード間の通信プログラムも Web 上で公開する。これによって、デジタル回路やプロセッサの設計演習などに関しては、HDL プログラムをベースに内部の変更や命令の追加ができる。また、計算機アーキテクチャやソフトウェアに力点を置いた講義等では、VGA ポートを持った FPGA ボードさえあれば、構成データを FPGA にダウンロードすれば使用できる。

基本的に多くの FPGA ボードに実装できるよう、上記のデータをボードごとに Web 上に用意するが、存在するすべてのボードに対してデータを準備するのは困難である。そこで、ユーザが所持する FPGA ボードへの移植を容易にするために、設計の階層でボード依存部と非依存部を明確に分離する。そして、ボード依存部は小規模化し、なるべく移植の負荷が減るようにする。

2.2 システム概要

図1にプロセッサ教育支援システムの構成を示す。FPGA に搭載される機構はすべて VHDL で記述されており、ボード非依存部 (TOP) とボード依存部 (BTOP) とに階層化されている。基本的には TOP は変更せずに、BTOP の記述を当該ボードのスイッチや LED、および、7セグメント LED や VGA 用 AD 変換器等に合わせて変更すれば良い。ただし、TOP には FPGA に依存するオンチップメモリがあり、それは使用 FPGA に合わせて変更する必要がある。

本システムは、ノイマン型プロセッサの動作原理を理解させることが目的である。さらに、安価な FPGA ボードを考慮し、図1のシステム全体が小規模な FPGA に

†九州工業大学大学院工学研究院

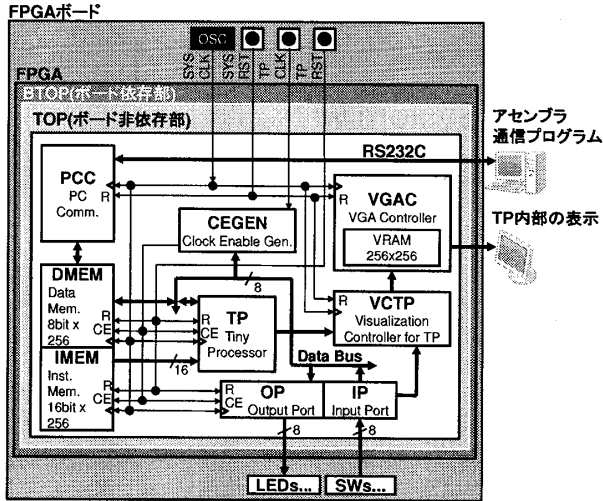


図1 教育支援システムの構成

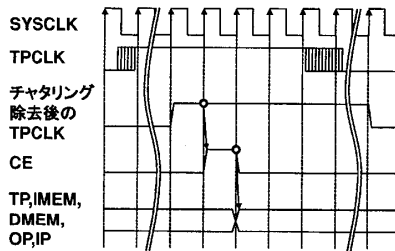


図2 ステップ実行の動作波形

も搭載できる必要がある。そこで、プロセッサ (TP) は割り込み機能を持たない単純な 8 ビットの逐次プロセッサとした。そして、内部構成は一般的な 3 バス構成で、命令は 16 ビットの固定長とした。

TP のレジスタ、データバス、メモリ制御信号、および、制御回路の状態は観測用に出力され、VCTP (Visualization Controller for TP) と VGAC (VGA controller) からなる動作可視化機構へと接続される。VCTP は画面上に文字列、レジスタの枠、マルチプレクサ、配線等の背景を書き出した後に、TP から出力されるデータと汎用入出力ポート (IP, OP) の内容を画面に表示し続ける。画面へは VRAM の内容がそのまま VGA 上に表示される。使用するディスプレイは、640x480 の表示が可能であればよく、小規模 FPGA への搭載を考慮して、表示画面は 256x256 で色は 2 色 (白黒) である。また、画面レイアウトの変更を簡単化するために、複雑な描画機能はすべて VGAC に持たせ (例えば、矩形や直線、文字列の描画など)、VCTP は、いくつかのコマンドとパラメータを VGAC へ渡し、描画を依頼する。つまり、VCTP の状態マシンにおいてコマンドの発行順を変えるだけで、画面のレイアウトが変更できる。

プロセッサの動作を目視するには、プロセッサをステップ実行できる必要がある。そのため、FPGA ボード上のスイッチ 1 個をステップクロック (TPCLK) の入力用に使用する。ステップクロックはボード非依存部内の CEGEN でチャタリングが除去され、ボードクロック (SYSCLK) に同期してワンショットパルス化される (図 2 の CE)。FPGA 内の回路はボードクロックに同期して動作しており、CE は TP に関係する回路 (TP, DMEM, IMEM, OP, IP) のクロック許可信号になる。よって、図 2 のように、CE が 1 になると TP 関係の回路が、ボード

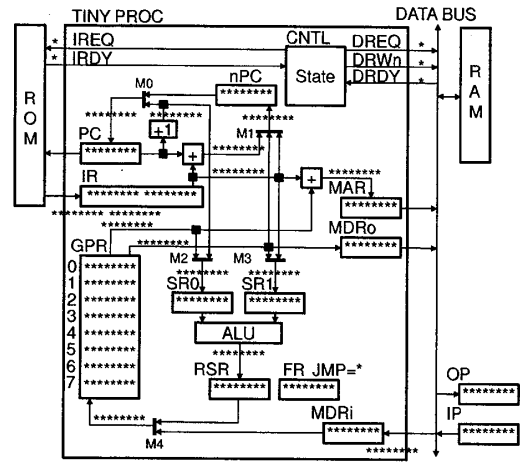


図3 画面構成 (TP のブロック図)

クロックに同期しながら 1 ステップだけ動作を進める。電源投入後やりセット後、TP はデフォルトでステップ実行である。TP のソフトウェアによってボードクロックとステップクロックを切り替えられるため、デバッグのブレーク機能と同様に、普段は高速動作し、ある部分のみステップ実行に切り替えることも可能である。リセットに関しては、TP 関係の回路のみをリセットできるように、それらのリセット (TPRST) と他回路のリセット (SYSRST) を分離した。

命令メモリ (IMEM) は 16 ビット × 256 語の容量であり、データメモリ (DMEM) は 8 ビット × 256 語の容量である。パーソナルコンピュータ (PC) は PC 通信部 (PCC) を介して IMEM と DMEM を読み書きする。DMEM と IMEM はデュアルポートメモリでかつ、PC 側のポートはボードクロックで動作するので PC からの読み書きと TP からの読み書きを同時に実行できる。

PC との通信には大部分の市販 FPGA ボードがサポートする RS232C のシリアル通信を用いる。PC からすると USB が一般的であるが、FPGA ボードによって USB のインタフェース IC が異なり、USB の通信ハードウェアは RS232C の通信ハードウェアよりも複雑なことからボード依存部の記述負荷が増大してしまう。また、PC が USB ポートしか持たない場合でも、USB から RS232C への変換ケーブルは市販されている。さらに、伝送データ量が最大でも命令で 512 バイト、データで 256 バイトであり、RS232C でも転送時間は問題にならないと推測される。

PC では perl による簡易アセンブラを用いて TP プログラムを作成する。そして、RS232C を介した一般的な通信プログラムによって、マシン語ファイルやデータファイルを DMEM や IMEM に対して読み書きする。

データバスには DMEM 以外に、汎用の入出力ポート (IP, OP) も接続される。ボード依存部を介して、これらの入出力ポートを当該 FPGA ボード上のスイッチや LED に接続できる。

2.3 TP の構成 (画面構成)

図 3 にディスプレイの画面構成 (TP のブロック図そのもの) を示す。画面には、レジスタの枠、データバス、マルチプレクサ等が描画され、レジスタとデータバス上の内容が 2 進数 (図 3 の*が 1 ビット) で表示される。

命令メモリ (ROM) に関して、命令要求信号 (IREQ),

Mnemonic	動作	Instruction Format																Flag Register (FR)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	G	GE	NE	E	OV	ZR	CY	
NOP		0	0	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0							
AND Rs, Rt	Rs <= Rs and Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	0	0	0				V		
OR Rs, Rt	Rs <= Rs or Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	0	1	0				V		
NOR Rs, Rt	Rs <= Rs nor Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	0	1	1				V		
XOR Rs, Rt	Rs <= Rs xor Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	1	0	0				V		
ADD Rs, Rt	Rs <= Rs + Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	1	0	1			V	V	V	
ADC Rs, Rt	Rs <= Rs + Rt + CY;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	0	1	1	0			V	V	V	
SUB Rs, Rt	Rs <= Rs - Rt;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	1	0	0	0			V	V	V	
SBB Rs, Rt	Rs <= Rs - Rt - CY;	0	0	s	s	s	s	t	t	t	t	*	*	*	*	1	0	0	1			V	V	V	
SRL Rs, Rt, SA	Rs <= 0 -> (Rt >> SA) -> CY;	0	0	s	s	s	s	t	t	t	t	a	a	a	a	1	0	1	0			V	V	V	
SLL Rs, Rt, SA	Rs <= CY <- (Rt << SA) <- 0;	0	0	s	s	s	s	t	t	t	t	a	a	a	a	1	0	1	1			V	V		
SRA Rs, Rt, SA	Rs <= Rt(7) -> (Rt >> SA) -> CY;	0	0	s	s	s	s	t	t	t	t	a	a	a	a	1	1	0	0			V	V	V	
CMP Rs, Rt, CC	if (Rs op(cc) Rt) FR(cc) <= 1; else FR(cc) <= 0;	0	0	s	s	s	s	t	t	t	t	c	c	c	c	*	1	1	0	1		V	V	V	V
CMPU Rs, Rt, CC	if (Rs op(cc) Rt) FR(cc) <= 1; else FR(cc) <= 0;	0	0	s	s	s	s	t	t	t	t	c	c	c	c	*	1	1	1	1		V	V	V	V
JR Rt	PC <= Rt;	0	0	*	*	*	*	t	t	t	t	*	*	*	*	1	1	1	0						
LW Rt, OFST[Rs]	Rt <= mem(OFST + Rs);	0	1	s	s	s	s	t	t	t	t	o	o	o	o	o	o	o	o						
SW Rt, OFST[Rs]	mem(OFST + Rs) <= Rt;	1	0	s	s	s	s	t	t	t	t	o	o	o	o	o	o	o	o						
ADI Rs, IMM	Rs <= Rs + IMM;	1	1	s	s	s	s	0	0	0	0	i	i	i	i	i	i	i	i			V	V	V	
ADIC Rs, IMM	Rs <= Rs + IMM + CY;	1	1	s	s	s	s	0	0	1	1	i	i	i	i	i	i	i	i			V	V	V	
JMP ADDR	PC <= ADDR;	1	1	*	*	*	*	1	0	0	0	a	a	a	a	a	a	a	a						
JAL ADDR	PC <= ADDR, R7 <= PC + 1;	1	1	*	*	*	*	1	0	1	0	a	a	a	a	a	a	a	a						
BRA CC, OFST	if (FR(cc)=1) PC <= PC+OFST; else PC <= PC + 1;	1	1	c	c	c	c	1	1	0	0	o	o	o	o	o	o	o	o						
BRAL CC, OFST	if (FR(cc)=1) { PC <= PC+OFST; R7 <= PC + 1; } else PC <= PC + 1;	1	1	c	c	c	c	1	1	1	0	o	o	o	o	o	o	o	o						

フラグ	動作	CC	分岐条件	CC	op(cc)
CY	演算の結果、キャリーが発生したら1, そうでなければ0	000	CY = 1のとき分岐	011	=
ZR	演算結果がゼロの場合に1, そうでなければ0	001	ZR = 1のとき分岐	100	!=
OV	演算結果がオーバーフローした場合に1, そうでなければ0	010	OV = 1のとき分岐	101	>=
EQ	CMP, CMPUのときにRs == Rtならば1, そうでなければ0	011	EQ = 1のとき分岐	110	>
NE	CMP, CMPUのときにRs != Rtならば1, そうでなければ0	100	NE = 1のとき分岐		
GE	CMP, CMPUのときにRs >= Rtならば1, そうでなければ0	101	GE = 1のとき分岐		
GT	CMP, CMPUのときにRs > Rtならば1, そうでなければ0	110	GT = 1のとき分岐		

図4 TPの命令フォーマット

命令レディ信号 (IRDY), プログラムカウンタ (PC), および、命令レジスタ (IR) が表示される。PC の入力 は次の命令 (PC+1) または、分岐先番地 (nPC) になる。nPC の入力 は、IR の即値と PC の加算値 (自己相対番地)、汎用レジスタ (レジスタ直接)、または、IR の即値 (絶対番地) となる。マルチプレクサの入力と出力が表示されるので、どのパスが選択されているのか目視できる。IR の下に表示される 16 ビットデータは命令メモリから出力された命令である。

ALU には 2 つの入力レジスタ (SR0, SR1) と 1 つの出力レジスタ (RSR) が接続される。ALU での演算結果に応じて設定されるフラグレジスタ (FR) も表示される。SR0 の入力 は、汎用レジスタか、次命令の番地 (後述するリンク付き分岐命令で使用) になる。SR1 の入力 は、汎用レジスタか、IR の即値になる。分岐成立時には、FR の横の表示が JMP=1 になり、不成立時は JMP=0 になる。

データバスに関しては、データ要求信号 (DREQ), データ読み書き信号 (DRWn), データレディ信号 (DRDY), メモリアドレスレジスタ (MAR), メモリデータ出力レジスタ (MDRo), メモリデータ入力レジスタ (MDRi) が表示される。MAR の入力 は IR の即値と汎用レジスタの内容との加算 (レジスタ相対番地) になる。汎用レジスタの内容が MDRo に入力され、データバスに出力される。データバスからの入力 は MDRi を介して GPR へ入力される。MDRi の下に表示される 8 ビットデータは、データバス上のデータである。

汎用レジスタ (GPR) は 8 個であり、2 つの出力ポートと 1 つの入力ポートを持つ。GPR への入力 は、ALU での演算結果 (RSR の内容)、または、メモリから読み出した値 (MDRi の内容) になる。

制御回路 (CNTL) には、TP の状態が表示される。状態は、命令フェッチ、命令デコード、演算実行、メモリアクセス、および、レジスタ書き戻しの 5 つであり、それぞれ、IF, DEC, EXE, MEM, WB と表示される。

汎用入出力ポート (OP, IP) の内容も表示され、FPGA ボードのスイッチからの入力 と LED 等への出力が 2 進数で直接確認できる。

2.4 TPの命令セット

図4にTPの命令セットを示す。命令は16ビットの固定長であり、上位2ビット(15,14ビット)で命令の大きなタイプを区別する。上位2ビットが"00"ならばレジスタ演算(R-Type), "01"から"10"ならばメモリアクセス(M-Type), "11"ならば即値演算(I-Type)である。R-Typeは、3~0ビット目でさらに細かく命令が分類され、I-Typeは10~8ビット目で分類される。

TPのフラグレジスタ(FR)は6ビットである。FRの2~0ビットが、各々、オーバーフロー(OV), ゼロ(ZR), キャリー(CY)である。6~3ビットは、比較命令によって設定されるフラグであり、それぞれ、より大きい(G), 以上(GE), 不一致(NE), 一致(E)である。Vは、命令の演算結果によってフラグが変化することを意味する。

図4の上から、R-TypeのAND~XORは論理演算である。ADDは算術加算、ADCはキャリー付きの算術

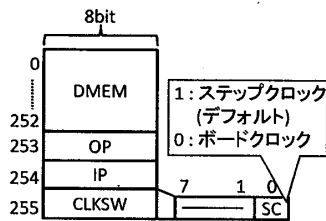


図5 TPからみたメモリマップ

加算である。SUBは算術減算、SBBはボロー付きの算術減算である。R-TypeのSRL~SRAはシフト命令である。SRLとSLLは論理右シフトと論理左シフト、SRAは算術右シフトであり、シフト量は7~5ビットで指定される。次のR-TypeのCMPとCMPUは比較命令であり、末尾のUは符号なしを意味する。比較方法は条件コード(CC)で指定される。CCはフラグレジスタのビット番号に対応し、CMPとCMPUではCCに6~3の値が使用される。JRは無条件ジャンプであり、指定されたレジスタの内容が分岐先番地になる。

LWとSWがM-Typeの命令であり、それぞれ、ロード命令とストア命令である。13~11ビットで指定されたレジスタの内容がベースアドレスとなり、7~0ビットの即値をオフセットとしてメモリがアクセスされる。

ADI~BRALがI-Typeの命令であり、その中のADIが即値加算、ADICがキャリー付きの即値加算である。JMPとJALは無条件分岐であり、それぞれ、7~0ビットの即値が分岐先番地になる。ただし、JALはリンク付きの分岐であり、JALの次命令の番地がR7レジスタに書き込まれる。BRAとBRALは条件分岐であり、CCで指定されたフラグが1ならば、分岐する。分岐先番地は7~0ビットの即値とPCの内容を加算した自己相対番地になる。BRALはJALと同様にリンク付きの分岐であり、分岐した場合、R7に次命令の番地が書き込まれる。

2.5 TPからみたメモリマップ

TPは実行するプログラムによって、データメモリや汎用入出力ポートを読み書きしたり、動作クロックをボードクロックからステップクロックに切り替えられる。これらの機能はメモリマップドIOによって実現され、TPから見たメモリマップを図5に示す。

図5のDMEM(データメモリ領域)は、図1のデータメモリ(DMEM)に対応し、任意の8ビットデータを読み書きできる。IPは8ビットの汎用入力ポートであり、FPGA外部から任意の8ビットデータを入力できる。OPは8ビットの汎用出力ポートである、FPGA外部に任意の8ビットデータを出力できる。CLKSWは、TPの動作をボードクロックか、ステップクロックに切り替えるためのIOマップドレジスタである。0ビット目を1にすることで、TPの動作クロックはステップクロックに、0にすると、ボードクロックになる。CLKSWのデフォルト値は1であり、電源投入後、または、リセット後、TPはステップクロックで動作する。

2.6 PCとの通信プロトコル

本教育支援システムは、PC上でアセンブラによってプログラムを作成し、命令コードをFPGA上の命令メモリ(図1のIMEM)へダウンロードする。必要ならば、データもFPGA上のデータメモリ(図1のDMEM)へダウンロードする。

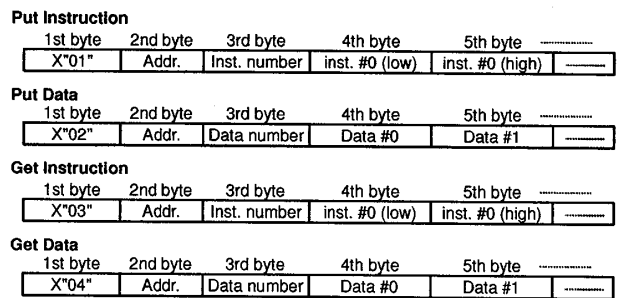


図6 シリアル通信プロトコル

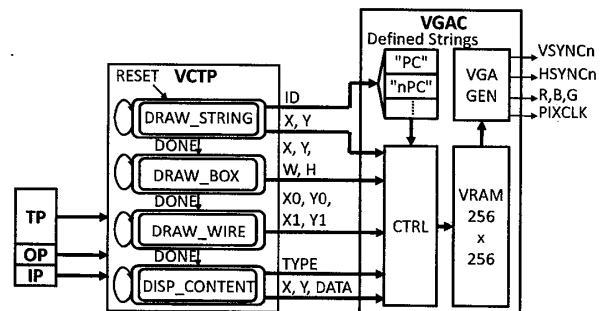


図7 動作可視化機構

IMEMは、ダウンロードが正しく行われたか確認できるように、PCから読み出すことができる。また、DMEMも、TPの演算結果を確認するためや、データが正しくダウンロードできたか確認するためにPCから読み出すことができる。

これらのPCからメモリに対する基本的な操作を図1のPCCがサポートする。PCとPCC間の通信プロトコルを図6に示す。最初の1バイト目がコマンドであり、続いて、読み書きするメモリの番地と転送する命令の数、または、データの数を送信される。その後、命令、または、データが転送される。命令は16ビット長なので、最初に下位バイトが、次に上位バイトが転送される。これらの単純なプロトコルを組み合わせると、PC上の通信プログラムが作られる。

3 VGAによる動作可視化機構

TPの内部状態をVGAに表示する機構を図7に示す。動作可視化機構は、図1のVCTPとVGACからなる。

VCTPはステートマシンであり、TPの内部信号と汎用入出力ポート(IPとOP)の内容表示をVGACに依頼する。VGACはVCTPからの依頼を受けて、VRAMに描画データを書き込む。それと同時に、VGAの表示タイミング信号を生成し、そのタイミングに合わせてVRAMの内容をディスプレイへ出力する。描画クロック(PIXCLK)も25MHzのクロック信号として生成される。

VCTPの状態群は、大きく4つに分類できる。図7の2重丸で囲まれた状態が各分類に対応する(2重丸は階層的に複数の状態からなることを意味する)。これらの分類は、文字列の表示(DRAW_STRING)、枠の表示(DRAW_BOX)、配線の表示(DRAW_WIRE)、および、データ内容の表示(DISP_CONTENT)である。DRAW_STRINGからDRAW_WIREまでは画面の背景であり、リセット時や電源投入後に一回だけ実行される。通常、VCTPはボードクロックに同期しながら、TP

表1 リソース量と動作速度 (SP3: Spartan3 Starter Kit[7], SP3E: Spartan3E Starter Kit[8], SP3A: Spartan3A Starter Kit[9], CYC2: Cyclone2 FPGA Starter Board[10], CYC3: Cyclone3 FPGA Starter Board[11])

Xilinx	Slices	LUTs	FFs	BRAMs	Speed
SP3	1918(99%)	3197 (83%)	1372 (36%)	10 (83%)	57.3MHz
SP3E	1975(42%)	3210 (34%)	1361 (15%)	10 (50%)	57.1MHz
SP3A	1972(33%)	3196 (26%)	1363 (12%)	10 (50%)	57.2MHz
Altera	LEs	FFs	MEM Bits	Speed	
CYC2	3609 (19%)	1368 (7%)	143360 (60%)	54.1MHz	
CYC3	3602 (15%)	1368 (6%)	71680 (12%)	79.2MHz	

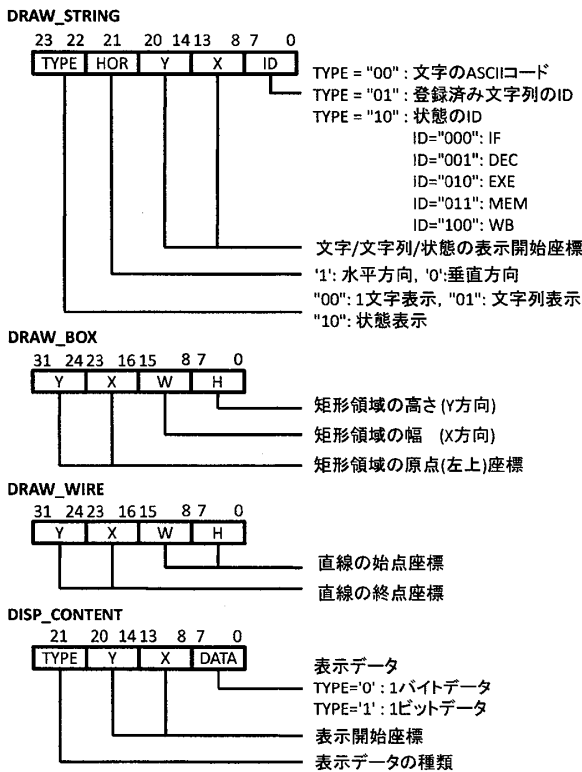


図8 VCTPのVGA表示コマンド

とは独立に DISP.CONTENT を実行し続けるので、TP のレジスタの内容、データバス、メモリ制御信号、および、制御回路の状態がディスプレイに表示され続ける。

画面構成の変更を容易ならしめるために描画制御の詳細を VGAC で隠蔽した。VCTP は画面描画にあたって図8のコマンドを VGAC に発行する。画面構成を変更するには、VCTP のステートマシンでコマンドの発行順と各コマンドのパラメータを変更すればよい。

図8の DRAW.STRING は1文字、4文字までの文字列、または、制御回路の状態を表示するコマンドである。文字フォントの大きさは4x4であり、画面上の文字座標は64x64となる。23~22ビット(TYPE)で表示の種類が指定され、21ビット(HOR)で表示する方向、続いて表示開始座標(Y, X)が指定される。表示内容は下位8ビット(ID)で指定される。1文字表示ならばIDはASCIIコードであり、文字列表示ならば文字列表の添え字になる。文字列表とは表示したい文字列をHDLコードにおいて表形式で定義したものである。状態表示の場合、IDは状態の識別子になる。

DRAW_BOX は矩形表示のコマンドである。上位バイトから、矩形の原点座標(Y, X)、矩形の幅(W)と高さ(H)が指定される。原点座標は矩形の左上である。

DRAW_WIRE は直線の表示コマンドである。上位バ

イトから、直線の終点座標(Y1, X1)、直線の始点座標(Y0, X0)が指定され、縦、横、斜めに直線を描画できる。

DISP.CONTENT は、TP の内部データを画面上に表示するコマンドである。21ビット(TYPE)で表示データの種類が指定され、'0'ならば8ビットデータ、'1'ならば1ビットデータである。表示文字は4x4の大きさなので、DRAW.STRINGと同様に、指定できる座標(Y, X)の大きさは64x64である。画面上に表示されるデータは最下位バイト(DATA)に入力される。VCTPは、TPからトップ階層に出力されている内部信号の種類に応じてTYPEとDATAを指定し、VGACに渡す。

4 実験と考察

4.1 実装実験

教育支援システムの設計データの汎用性を評価するために、いくつかの市販ボード [7, 8, 9, 10, 11] に実装した。市販ボードは、Xilinx社のボードが3枚、Altera社のボードが2枚である。前者のFPGAは、各々、Spartan3, Spartan3E, および、Spartan3Aであり、後者のFPGAは、各々、Cyclone II, Cyclone IIIである。使用したCADツールは、Xilinx社のISE10.1.03iと、Altera社のQuartus II 9.0である。HDLプログラムに関して、ボードごとにボード依存部の記述を変更した。さらに、Sparan3*とCyclone*でオンチップメモリを変更する必要があり、CADツールのIP(Intellectual Property)生成ツールを使って同じ仕様のオンチップメモリ(デュアルポートRAM)を生成した。なお、これら市販ボードのボードクロックは50MHzである。

市販ボードへの実装結果を表1に示す。SP3はSpartan3, SP3EはSpartan3E, SP3AはSpartan3Aを搭載したボードであり、CYC2はCyclone II, CYC3はCyclone IIIを搭載したボードである。

本教育支援システムは実験で用いたすべてのボードに搭載できた。動作速度は50MHz以上を達成しており、ボードクロックをそのまま使用できる。また、SP3以外は、ロジック資源(LUTやLE)に余裕があり、割り込み等のその他の機構も盛り込むことができる。さらに、SP3以外はVRAMに使用できるオンチップメモリに余裕があるため、画面表示をより高機能化しうる。例えば、VRAMのビットサイズを増やしてVGA上の表示をカラー化したり、表示エリアを拡張したりできる。

図9に使用したロジック資源の各モジュールごとの内訳とコード量(行数)の内訳を示す。グラフの凡例は図1に示した各モジュールに対応する。ロジック資源の内訳は各FPGAの最大容量を1として正規化されており、コード量は全体の行数を1として正規化されている。

ロジック資源の内訳を見ると、SP3~SP3A間の傾向は同じであり、CYC2とCYC3間でも同様である。これ

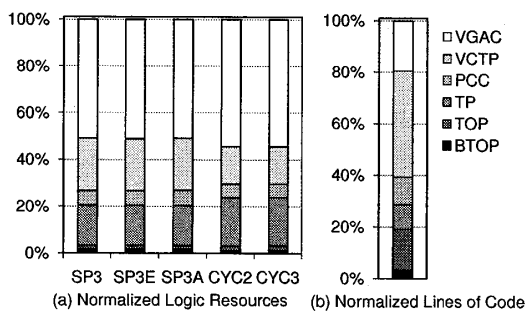


図9 リソース量とコード量の内訳

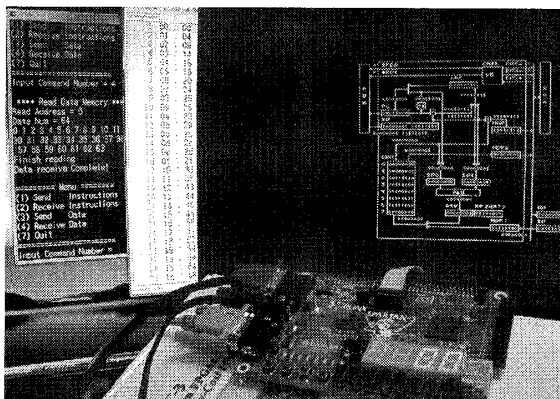


図10 Spartan3 Starter Kitにおける教育支援システムの動作例(ランダムデータに対するバブルソート)

は、搭載されたFPGAのアーキテクチャが類似していることによる。また、ボード全体にわたって、ボード依存部(BTOP)のハードウェア量は小さい(1.4%以下)。さらに、ボード依存部のコード量は全体の3.3%以下と小さく、実際にボード間での移植の負荷は少なかった。

また、画面の描画を指示するVCTPとTPが要したロジック資源量はほぼ同じであり、コード量に関しては、VCTPがTPの約4.3倍を要した。画面描画の簡単化を目的に、描画制御の詳細をコマンドとパラメタによってVCTPに対して隠べいしたが、画面構成の拡張等を考えた場合、変更の負荷が大きくなると考えられる。VCTPとTPの回路規模は同等であり、これ以上、描画の拡張をするのであれば、VCTPをプロセッサ化しソフトウェアで画面構成を変更できるようにすべきであろう。

4.2 動作確認

前節で使用したFPGAボード上で実際にプログラムを動作させた。その様子を図10に示す。動作させたプログラムは、各命令の動作検証プログラム(ステップ実行による目視確認)、および、配列要素の総和と、符号付きと符号なし数に対するバブルソートである。すべてのFPGAボードでこれらのプログラムが正しく動作することを確認した。

5 むすび

私たちの身の回りに存在するあらゆる電子機器にはプロセッサが搭載されており、工学教育上、プロセッサの理解は重要である。シミュレーションによるプロセッサ教育支援システムは、汎用性が高い反面、プロセッサの動作が画面上に表示される仮想空間のできごとであり、初心者がノイマン型プロセッサの動作原理を直感的に理解し難い。一方、FPGAボードを使用した教育支援シス

テムでは、実際にハードウェアに触れさせ、自らそれを動かすことを通し、高いモチベーションを保ちながら動作原理の理解を支援する。ただし、専用ボードの入手が困難であったり、ボード上での動作可視化能力に限界がある。

そこで、様々な市販FPGAボードに搭載可能で、かつ、VGAによる動作可視化機構を備えたプロセッサ教育支援システムを開発した。本システムを講義や自習に広く使用できるように、設計データをWebから自由に入手かつ変更できるようにした。いくつかの市販FPGAボードに教育支援システムを実装したところ、ボード間で設計データを移植する負荷は小さく、それらすべてのボードで正しく動作することを確認した。

今後の課題としては、回路資源に余裕のあるFPGAを搭載したボード向けに表示機能を拡張することと、同時に画面をソフトウェアで構成できるようにすることが挙げられる。また、DMAなどの周辺機器との協調動作(バス競合や割り込み)も可視化できるようにする。さらに、文献[1, 2]に示されるような高機能なソフトウェア教育環境と本ハードウェアとの連携も予定している。

参考文献

- [1] Mihyar Hesson. Computer Simulator: An Educational Tool for Computer Architecture. *American Journal of Applied Sciences*, Vol. 3, No. 11, pp. 2114-2121, 2006.
- [2] 今井慈郎, 金子敬一, 中川正樹. 計算機アーキテクチャ教育支援システムの開発と協調学習への適用. *信学論*, Vol. J91-D, No. 2, pp. 188-199, 2008.
- [3] 重村哲至, 古川達也, 相知政司, 林敏浩. コンソールパネルを持つ機械語教育用マイコンの開発と授業への応用. *情処学論*, Vol. 48, No. 9, pp. 3318-3327, 2007.
- [4] 末吉敏則, 久我守弘, 柴村英智. KITE マイクロプロセッサによる計算機工学教育支援システム. *情処学論*, Vol. J84-D-I, No. 6, pp. 917-926, 2001.
- [5] Kasim M. Al-Aubidy. Teaching Computer Organization and Architecture Using Simulation and FPGA Applications. *Journal of Computer Science*, Vol. 3, No. 8, pp. 624-632, 2007.
- [6] Koji Nakano, Kensuke Kawakami, Koji Shigemoto, Yuki Kamada, and Yasuaki Ito. A tiny processing system for education and small embedded systems on the fpgas. In *Proc. of 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008.
- [7] Xilinx. *Spartan-3 Starter Kit Board User Guide UG130*. Xilinx, 2005.
- [8] Xilinx. *Spartan-3E Starter Kit Board User Guide UG230*. Xilinx, 2006.
- [9] Xilinx. *Spartan-3A/3AN Starter Kit Board User Guide UG334*. Xilinx, 2007.
- [10] Altera. *Cyclone II FPGA Starter Kit Board User Guide*. Altera, 2006.
- [11] Altera. *Cyclone III FPGA Starter Board Reference Manual*. Altera, 2007.