鹿児島女子短期大学紀要
第48号（2013）81～87頁

ホームページ上への仮想 CPU の具現化

Implement Virtual-CPU on Homepage

瀬　戸　博　幸
Hiroyuki Seto

キーワード：仮想コンピュータ，具現化，ホームページ，JavaScript，教材

１．はじめに

筆者は『コンピュータ基礎論』の教材として，仮想的なコンピュータを Microsoft Access のアプリケーションとして作成し，実際に講義で使用してきた[(1)]。本稿では，このコンピュータを，より広汎に使用できるようホームページ上に実現したので報告する．

２．実現した仮想的なコンピュータ

教材用仮想コンピュータとしてはクヌースの MIX などがある[(2)]が，教養的な科目の教材として使うにはコンピュータの構成が複雑であり命令数も多すぎる．そこで，CPU の基本的構成やメモリーとの関係，および，機械語によるプログラミングを通して，ストアードプログラム方式の概要を理解させる目的で，図１のように仮想的なコンピュータを設計し Microsoft Access のアプリケーション（以下 VCPU1 と記述する）として実現し，教材として使用してきた.[(1)]

Microsoft Access のアプリケーションとして VCPU1 を実現したのは，

- メモリーとしてデータベースが利用できる
- アプリケーションの入出力を，フォームを使って GUI として簡単に設計できる
- CPU の動作を VBA マクロでプログラムとして実現できる

以上の理由であるが，Microsoft Access のライセンスのないコンピュータでは実行できないので，その環境が無い学生には提供できなかった．また，VCPU1 を配布し実行させるまでには，少し時間を要した．

そこで今回，図２のようにホームページ上に実現し（以下 VCPU2 と記述する），配布も含めてより簡単に実行できるようにした．

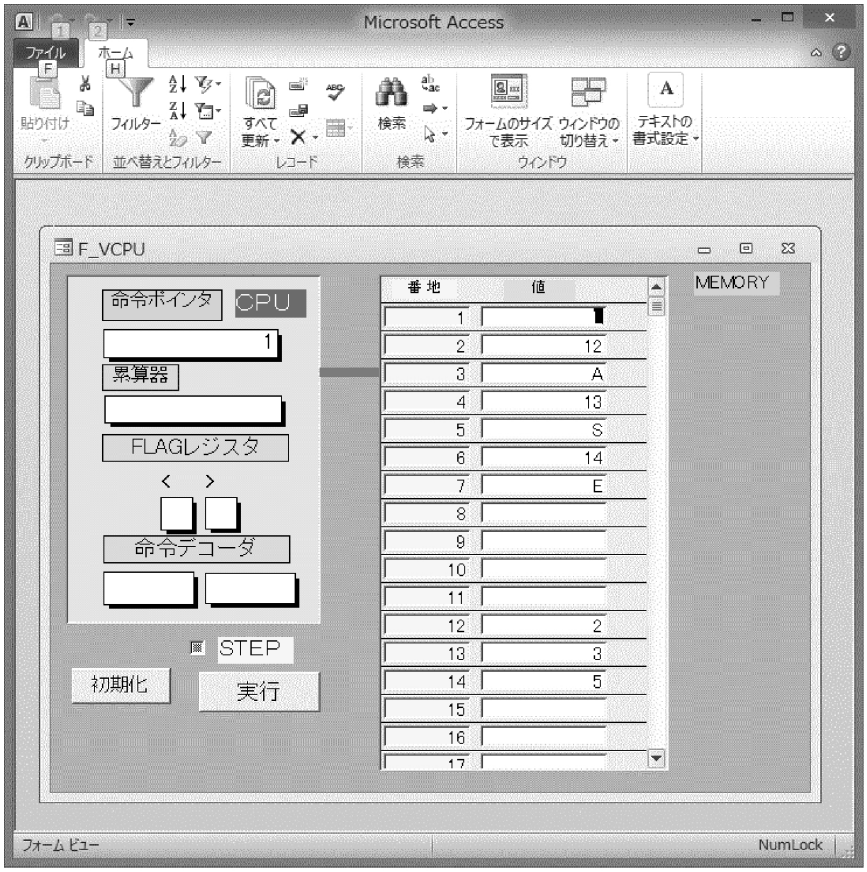

図１．Microsoft Access 上で稼働する VCPU1 の実行画面

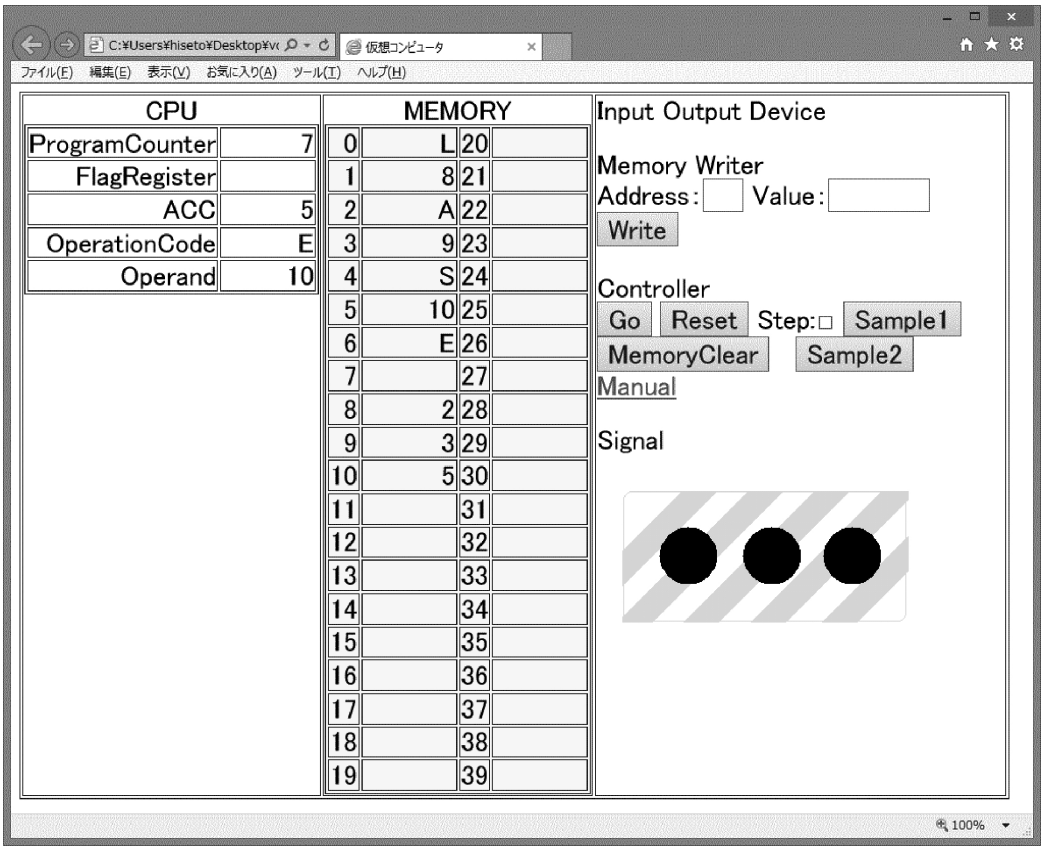

図２．ホームページ上に実現した VCPU2 の実行画面

３．VCPU2 の構成

VCPU2 は大きく CPU，Memory および Input-Output-Device に分かれている．

CPU および Memory は HTML の TABLE タグで実現した．

CPU には ProgramCounter，FlagRegister，アキュムレータ（ACC）および OperationCode と Operand が１つずつ存在するだけの最小限の構成にした．一般にコンピュータの命令（Instruction）は CPU の動作を示す OperationCode と，その動作の値となる Operand に分けられる．VCPU2 は実行処理（Execution）を簡単にするために OperationCode と Operand を，その順に直列する２つのメモリーセルに配置することとした．また，数値は10進数で表すこととした．さらに，実際のコンピュータでは機械語もデータも２進数的内部表現（電気信号）であるが，VCPU2 では OperationCode もアセンブリ言語のニーモニク表現を直接記述し，そのまま実行できるようにして，学生がプログラムに専念できるようにしている．

Memory は Address と Value の組が１次元的に配列されるものであるが，画面をスクロールしなくてもプログラム全体が見えるように２次元的に配置した．これは VCPU1 を使用した結果，ステップ数の多いプログラムで全体が見渡せない不便さを学生が感じていたので，VCPU2 で改善したものである．命令実行を実現するエンジン（VCPU2 を構成するプログラム）は少し複雑になるが全体を見渡せる理解のしやすさを重視した．

Input Output Devise 部は，使用する人間とコンピュータをつなぐマンマシンインターフェースであり，VCPU2 では，『Memory Writer』，『Controller』，および『Signal』を置いた．

VCPU1 ではメモリーセルも CPU 内の各レジスタも，Microsoft Access のフォームを用いて作成したので，直接クリックして書き込むことができたのであるが，VCPU2 では TABLE タグで実現しているので，クリックして直接入力することはできない．そこで『Memory Writer』と名付けた入力装置として，INPUT タグのテキストおよびボタン属性で GUI を用意し，TABLE への書き込み処理は JavaScript で関数を組んで実現した．結果として，実際のコンピュータの操作により近づいたと言える．

『Controller』は VCPU2 のプログラムを実行する際の制御装置である．VCPU1 では CPU の各レジスタを初期化する初期化ボタンとプログラム実行を開始する実行ボタン，および，命令を１ステップずつ実行させるための Step チェックボックスしか実装していなかったが，VCPU2 ではさらに Memory-Clear ボタンおよび Sample1 ボタンと Sample2 ボタンを追加した．現実のコンピュータでは Reset ボタンと Step スイッチぐらいしかなく，Reset ボタンを押して離すタイミングで割り込みが掛り，CPU がメモリ上の機械語プログラムを実行し，もし Step スイッチがオンであれば１つの命令を実行する毎に停止するデバッグモードがオプション的に付加されている程度であるが，VCPU1 も VCPU2 も Reset ボタンには CPU の内容を初期化する機能，Go ボタンにはプログラムの実行を開始する機能を割り当てた．学生が教材としてプログラムを作成し，確認し，意識して実行できるよう配慮したものである．また，プログラムやデータの記憶に使わないメモリーの内容はどのような値が入っていても，プログラムの実行に差し支えはないが，プログラムの視認性をよくするために，すべてのメモリーを空にする Memory-Clear ボタンを用意した．さらに，プログラムを

作成する技能を，まだ身につけてない学生でも，まず手始めに VCPU2 の動作を視認できるよう，サンプルプログラムをメモリーに書き込む Sample1 ボタンと Sample2 ボタンを用意した．Sample1 は２＋３の簡単な計算をするプログラムであり，Sample2 は『Signal』と名付けた信号機を制御するための，VCPU2 が実行可能なすべての OperationCode を使った，サンプルプログラムである．

『Signal』はメモリーの39番地の値を0.5秒間隔で監視し，その値が１ならば青ランプ，２なら黄色ランプ，３ならば赤ランプが点灯し，それ以外の値であれば全ランプ消灯となる出力装置である．その動作は，JavaScript の関数で記述しており，VCPU2 ホームページ表示開始と同時に監視活動を開始する．VCPU1 では特に出力装置を用意しなかったが，学生が VCPU2 をより身近に感じて，プログラムを作成することに意欲を燃やすことを期待して用意した．

さらに，VCPU1 では操作手引書を Word の文書として用意したが，VCPU2 ではホームページの特性を生かして，ホームページとして操作手引書を用意し，Controller の下に，そのページへのリンクを作った．

タブブラウザでは VCPU2 を操作している途中でも，クリックひとつで操作手引書を別タブとして呼び出せ，タブを切り替えるだけで，簡単に参照できるようになった．

４．VCPU2 の命令実行手順

本節では VCPU2 がどのように機械語命令を実行するかを説明する．なお，VCPU1 および VCPU2 は同一の命令体系をもっている．

VCPU2 のプログラムとは，Memory に OperationCode と Operand の組が連続して記述されたものである．VCPU2 の命令実行サイクルは Go ボタンをクリックすることで起動し，次の①から⑥の順序で，命令を読み込み，実行するものである．

① ProgramCounter が示す Address の内容を OperationCode に読み込む．
② ProgramCounter の値を１増やす．
③ OperationCode が Operand を必要とするか，判定し必要なければ⑥に進む．
④ ProgramCounter が示す Address の内容を Operand に読み込む
⑤ ProgramCounter の値を１増やす．
⑥ OperationCode を実行する．

VCPU2 では上記手順を JavaScript の関数として記述し，停止条件を判定して停止でなければ，setTimeout 関数により上記関数を再帰的に呼び出すことで具現化している．

停止条件とは

- 実行した OperationCode が E または e である
- 実行できない OperationCode である
- Controller の Step チェックボックスがチェックされている

であり，以上の場合に再帰的呼び出しをしないので，命令実行サイクルが停止する．

VCPU2 で実行可能な OperationCode の一覧を図３に示す.

Instruction Set

OperationCode	動作説明
L	Operandで示すAddressのMemoryの内容をACCに複写する
S	Operandで示すAddressのMemoryにACCの内容を複写する
A	Operandで示すAddressのMemoryの内容とACCの内容を加算し結果をACCに残す
E	「停止します」と表示し停止する
J	OperandをProgramCounterに複写する
C	Operandで示すAddressのMemoryの内容とACCの内容を比較し ACC＜Memoryなら1 ACC＞Memoryなら2 Memory＝ACCなら0をFlagRegisterに設定する
JE	もしFlagRegisterが0ならばJと同じ さもなければ何もしない

図３．VCPU2 で実行可能な命令一覧

実行可能な OperationCode は７つと非常に少なく，なおかつ，単純であるので理解しやすい．さらに，L,S,A のようにアセンブリ言語のニーモニック表現を，機械語としてそのままメモリー上に記述して実行できるので翻訳の知識も必要なく，簡単な操作を覚えるだけですぐにプログラムに専念できる.

それぞれの命令は JavaScript の関数として記述し，上記実行サイクルの⑥のところで，その関数を呼び出している.

よって，JavaScript の知識があれば，新しい OperationCode が必要な場合，それに応じた関数を記述し，実行サイクルに組み込めば，簡単に実現できる.

なお，入力上の便宜を考えて OperationCode は大文字でも小文字でも実行できるように配慮している.

５．考察

様々な講義で学生がホームページ作成の演習などに使用している学生用ファイルサーバに VCPU2 を置き，トップページからリンクを張って VCPU2 を使用してみた.

当然，URL を指示する簡単な説明で学生は VCPU2 を学生個々のコンピュータのブラウザ上に起動できた．VCPU1 ではプログラムのダウンロードなどの経験の少ない学生に，ダウンロードやプログラムの起動の説明をしていたのに比べ，非常に簡単に VCPU2 そのものの説明に進めた.

VCPU1 ではデータベースがメモリーになっているので，図１のようにあらかじめ簡単なプログラムをデータベース（VCPU1 のメモリー）に仕込んでおけば，VCPU1 を起動して実行ボタンをクリックするだけで VCPU1 のプログラムを実行できた．同様に VCPU2 では Sample1 ボタンをクリックし，Go ボタンをクリックするだけで VCPU2 の動作を確認させることができる．さらに，MemoryClear ボタン，Reset ボタン，Sample2 ボタン，そして最後に Go ボタンをクリックするだけで，めまぐるしく変化する CPU の内容とともに，Signal が青ランプ，黄色ランプ，赤ランプの順

に点灯して停止するSample2プログラムの動作をVCPU2上で学生に見せることができる．

Sample1プログラムおよびSample2プログラムの実行を何度か繰り返すだけで，学生はストアードプログラム方式の概念を理解でき，さらに，VCPU2の機械語のプログラムにも興味を示すようになった．

こうなればMemoryWriterを使って，地道に学生自身の考えたプログラムをメモリーに書き込み，実行する作業も苦ではなくなるようである．そして，パソコンだけでなく様々な機器に組み込まれたコンピュータのイメージも実感できるようである．

次にVCPU2を実行するブラウザに着目してみる．

当初，VCPU2を実行するブラウザとしてInternet Explorer向けに開発を進めた．現在，図２のようにWindows8に装備されているInternet Explorer10でもVCPU2を実行できることを確認している．

考察の冒頭で述べたように，学生用ファイルサーバにVCPU2を置いたので，様々なブラウザを用いてVCPU2を起動した結果，Opera，Google Chrome，および，Androidタブレットのブラウザでも VCPU2が動作することを確認した．JavaScriptで記述しているので当然と言えるが，Google Chrome，および，Androidタブレットは Internet Explorerと全く同様の動作をしたが，Operaでは Memory Clearボタンの動作，つまり，TABLEタグの各TDタグに空白を書き込む処理に違いが見られた．Internet Explorer等では空白を１文字として処理したが，Operaでは空白をブランク文字として認識したようで，ボーダーの内側の線がない状態（へこみが無い状態）になった．しかし，Memory Writerで文字を書き込むと正常に書き込め，Sampleプログラムも正常に動作したのでOperaでVCPU2は動くと判定している．

なお，現状で，FireFoxはTABLEタグへの書き込みが全く動作せず図２の０から39のアドレス番号すら表示されない状態で止まってしまう．JavaScriptのTABLEへの書き込み命令の記述形式の問題だと考えており，調査中である．

最後に，VCPU2の今後について考えてみる．

図２のSignalを実現した関数はVCPU2をブラウザで表示した時点から0.5秒間隔でメモリの39番地を監視し続けている．また，無限ループとなるプログラムを書きGoボタンで実行している状態でもMemory Writerでメモリーに値を書き込むことが出来る．つまり，JavaScriptはタイムシェアリングな関数を簡単に記述することが出来る．よってVCPU2に割り込み処理のシミュレーション機能を持たせることも可能だと考えている．また，Dart[(3)]などJavaScriptを簡単に作り出す開発環境も出てきているので着目したい．さらに，VCPU2をJavaScriptで実現したことにより，JavaScriptを通してダイナミックなホームページを学習する科目と位置付けている「ウェブプログラミング演習」の教材としても使えると考えている．

６．まとめ

講義に使用する教材として作成したMicrosoft Accessのアプリである仮想CPUを，JavaScriptを使ってホームページ上に，より使いやすい形に再構築した．さらに今後の使用について考察した．

引用文献

１）「電子計算機の教育についての一方法」，瀬戸博幸，鹿児島女子短期大学　紀要　第34号

２）*The Art of Computer Programming Volume 1 Fundamental Algorithms Third Edition*, Donald E.Knuth, Addison-Wesley, 1997

３）*What is Dart?*, Kathy Walrath Seth Ladd, O'Reilly, 2012

（2012年12月７日　受理）