ハードウェア動作記述言語：ALHARD（3）
-- シミュレーションインタフェース --

2U-5

鶴 薫* 加藤 幸男*　杉本 明** 小島 泰三** 阿部 茂**
*三菱電機（株）コンピュータ製作所　**同中央研究所

1．はじめに

ハードウェアの仕様・動作レベルのシミュレーションを目的として、ハードウェア動作記述言語 Alhard の開発を行った[1)2)]。本稿では、シミュレーションにおけるユーザインタフェースについて述べる。

2．コマンドインタフェース

表1にシミュレーションコマンドをまとめる。コマンドは次の3種類に分かれる。

2．1 共通コマンド

コマンドファイルの読み込みや、ログファイルの設定など入出力関係の機能と、データの表示形式やシミュレーション中の警告出力などを制御する環境変数の設定などが行われる。

2．2 シミュレーションコマンド

シミュレーションコマンドは、主としてマイクロプログラムのデバッグ用に作成したものである。start や step コマンドは、シミュレーションターゲットのクラスの同一名のメソッドを起動し、シミュレーションのスタート状態の設定や実行が行われる。

Alhard の演算データには、4ビット幅のフラッグが付け加えられている。そして、各ビットをシミュレーション中のデータの追跡用マーカとして使用できるよう、演算においては、オペランドのフラッグの論理和が取られる。また、その内の1ビットを未定義値の表現に用いている。init コマンドは、-init で指定したリソース以外のデータには、すべて未定義値のビットを立てる。その後、すべてのリソースの init メソッドを実行し初期化する。このような機構により、未定義値のレジスタへの代入などの際には警告メッセージが出力される。red, green, blue コマンドは eval と同様にリソースの参照更新をおこなうが、フラッグの残りの3ビットをユーザ指定のマーカとして設定する。

trace コマンドや break コマンドは、Alhard のデーモン機能を利用して、リソースの参照時や更新時に内容表示やシミュレーションの中断を行う。例えば、
trace PC:A(12,8) >= 4 のコマンドは、レジスタ A が更新された場合に、その12ビットから8ビットまでの値が4以上の場合にレジスタ PC を表示する。また、
break =CM[100,120] とすると、100番地から120番地の範囲に含まれるアドレスにより制御メモリCMの読出しが行われた場合に、シミュレーションを中断する。

表1 シミュレータコマンド

共通コマンド	set	環境変数の参照と設定
	do	コマンドファイルの起動
	log	ログファイルの設定
	help	ヘルプ
	quit	終了
	visual	ビジュアルモードへの移行
シミュレーションコマンド	load	マイクロプログラムのロード
	init	初期化
	-init	初期化の禁止設定
	reset	リセット
	step	1ステップの実行
	run	継続実行
	start	スタートメソッドの呼び出し
	eval	リソースの参照更新
	red,green,blue	リソースの色設定
	trace	トレースの設定
	-trace	トレースの解除
	break	ブレークの設定
	-break	ブレークの解除
ハードウェア記述デバッグコマンド	trap	ブレークポイントの設定
	-trap	ブレークポイントの解除
	notify	メソッドトレースの設定
	-notify	メソッドトレースの解除
	continue	ブレークポイントからの続行
	next	1ステートメントの実行
	where	スタックフレームの表示
	var	メソッド変数の参照更新

ALHARD: A Behavioral Hardware Description Language (3)
Kaoru TSURU, Sachio KATOU, Akira SUGIMOTO, Taizo KOJIMA, Shigeru ABE
MITSUBISHI ELECTRIC CORPORATION

2．3　ハードウェア記述デバッグ用コマンド

Alhard によるハードウェア記述自体をデバッグするためのコマンドである。notify や trap はメソッドのトレースやエントリポイントでのブレーク設定を行う。また、next は Alhard 記述を1ステートメント実行する。where はメソッド呼出しのスタックフレームの表示を行い、var によりメソッドの一時変数の参照更新を行うことができる。

3．ビジュアルインタフェース

アイコンやポップアップメニュによる画面インタフェースである。レジスタやバスは、データのユーザ指定のマーカを色として使用し、現在値を表示する。従って、シミュレーションの進行をアニメーションとして見ることができる。また、画面表示を前の状態に戻らせるプレーバック機能も用意されている。戻らせることが可能なステップ数は、あらかじめユーザが設定する。

表2にシミュレーション対象のリソースを画面上に構成するため用意したアイコンの種類を示す。また、図1にアイコン配置 a) のC言語による記述例 b) を示す。図 b) の1行目の data_box はレジスタに対応するアイコンを生成する。図中のパラメータの self は付加する親アイコン、8 は表示文字列長、MARK はマーカ表示を、std は使用する文字フォントを表す。また、DEV(AREG) は対応するレジスタである。アイコンは、対応するレジスタに更新デーモンを設定し、値をモニタする。3行目の align は配置のためのマクロである。 4行目のareg とbreg を結ぶラインは、ハードウェア記述において対応するバスがなく、 AREG に対して参照デーモンを設定することによりデータの流れをモニタしている。

4．おわりに

本稿では、 Alhard シミュレーションシステムのユーザインタフェースについて報告した。現在、アイコンインタフェース記述用の問題向き言語の設計をおこなっており、 Alhard との組合せにより、ビジュアルシミュレータ構築をより容易にしていきたいと考えている。

参考文献

1) 杉本，小島，阿部，鶴，加藤：ハードウェア動作記述言語 ALHARD(1)，情処37全大(1988).

2) 小島，杉本，阿部，鶴，加藤：ハードウェア動作記述言語 ALHARD(2)，情処37全大(1988).

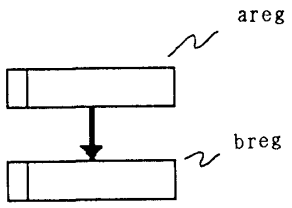

図1-a　配置例

表2　アイコンの種類

Text	名前表示用アイコン
DataLine	バス対応アイコン
DataBox PairDataBox	レジスタ対応アイコン 2重データボックス
MemorySlot AddressSlot PairSlot MemoryBox MemoryTable	メモリ内のスロットに対応 メモリスロットのアドレス表示 アドレスとスロットの組合せ メモリの参照・更新表示 メモリ内容の一覧
ResourceBox ResourcePanel	リソースの集まり（長方形） リソースの集まり（任意図形）

```
areg = data_box(self,8,MARK,std,DEV(AREG));                              - 1
breg = data_box(self,8,MARK,std,DEV(BREG));                              - 2
align(breg,TopLeft,pt_down(BottomLeft(areg),30));                        - 3
v_data_line(self,BottomCenter(areg),Top(breg),width,ARROW,DEV(AREG),GET); - 4
```

図1-b　記述例

図1　アイコン配置の記述例