

JavaRock-Thrash コード変換説明

1 Java から Verilog HDL へのコード変換規約

本章では，Java の記述から変換される Verilog HDL のコード例を紹介する．

1.1 クラス，メソッド，変数宣言の変換

Java ソースファイルのクラス，メソッド，変数宣言は，モジュールやポート，レジスタ，Block RAM などに変換される．以下でその変換例を示す．

1.1.1 クラス

Java のクラスは，Verilog HDL のモジュールに変換される．モジュールは，同期式順序回路であり，クロック (clk) とリセット信号 (reset) を持つ．図 1 にクラスから作成されるモジュールの例を示す．

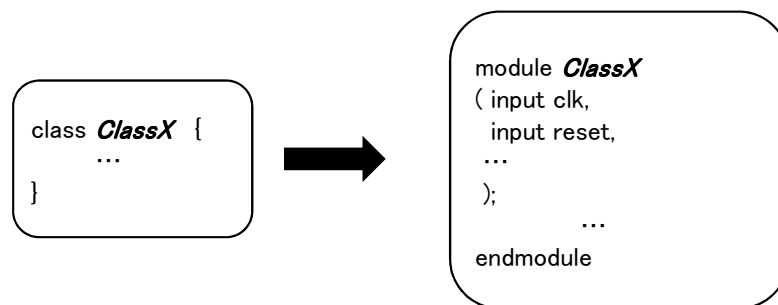


図 1: クラス → モジュール

1.1.2 メソッド

JavaRock Verilog では，各メソッド毎に処理の開始を命令する入力ポート（メソッド名_req）と，メソッドが処理中かどうかを示す出力ポート（メソッド名_busy）が存在する．また，メソッドの戻り値がプリミティブ型の場合，戻り値の出力ポート（メソッド名_return）が存在する．図 2 にメソッドから作成されるポート例を示す．

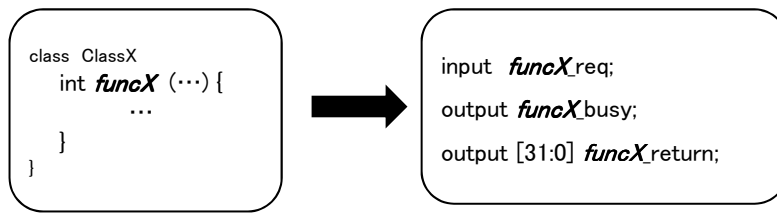


図 2: メソッド → ポート

1.1.3 パラメータリストの変数

パラメータリストに変数を宣言すると、それぞれの変数に対応した入出力信号が作成される。変数がプリミティブ型の場合、入力ポート（メソッド名_変数名）が宣言され、メソッド開始時にこのポートの値がモジュール内部のレジスタに格納される。一方、変数が配列型の場合、モジュール外部にある Block RAM 制御用の 4 つの入出力信号（メソッド名_変数名_addr, メソッド名_変数名_datain, メソッド名_変数名_dataout, メソッド名_変数名_r_w）が宣言される。図 3 にパラメータリストから作成されるポート例を示す。

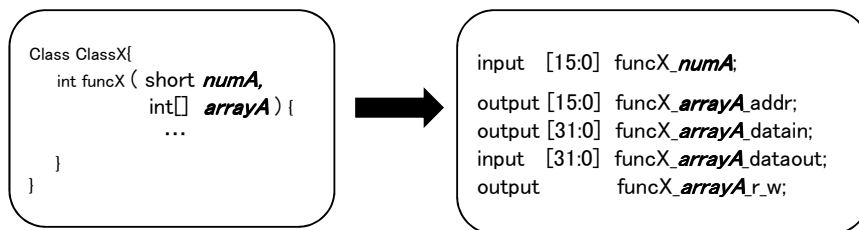


図 3: パラメータリスト → ポート

図 3 で示した Java ソースコードから作成されるモジュールに Block RAM を接続したブロック図を図 4 に示す。

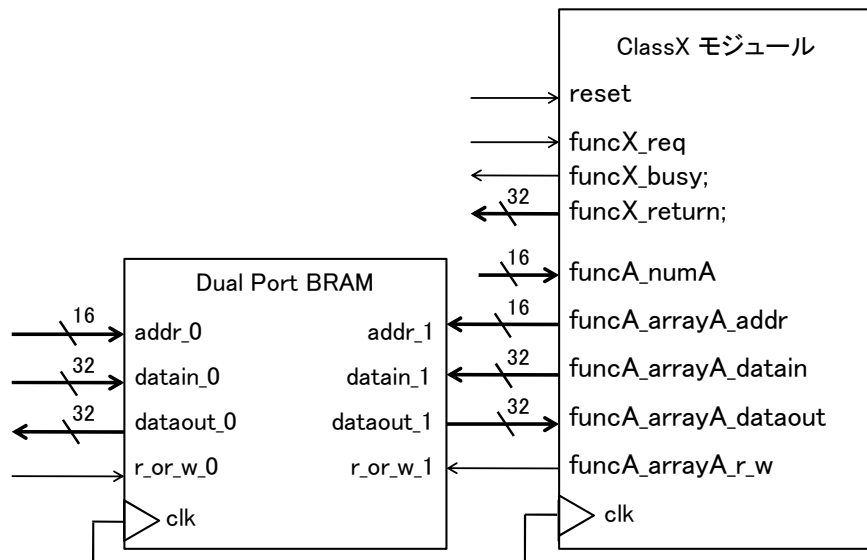


図 4: ClassX モジュールと外部 Block RAM の接続図

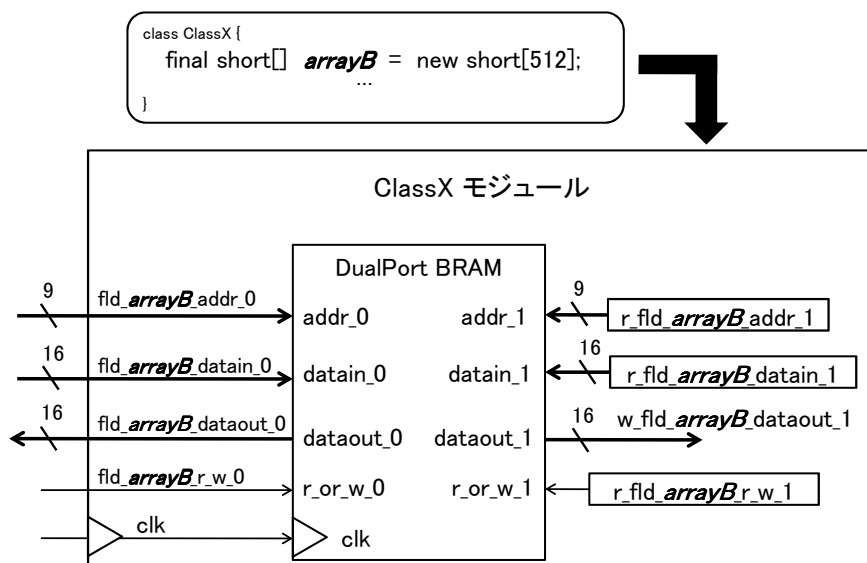


図 5: ClassX モジュールと内部 Block RAM の接続図

1.1.4 フィールドに宣言した配列型変数

フィールドに final 指定した配列変数を宣言すると、モジュール内部にデュアルポート Block RAM が作成される。この Block RAM の片方のポートは、モジュールの入出力ポートにつながっており、モジュール外部からデータの入出力が可能となっている。もう片方のポートには、モジュール内部でデータを入出力するためのレジスタやワイヤが接続される。図 5 にフィールドに宣言した配列から作成される Block RAM とモジュールの接続図を示す。fld から始まる名前の信号は、モジュールの入出力ポートである。r.fld, w.fld から始まる名前の信号は、モジュール内部で宣言した Block RAM の入出力用レジスタ及びワイヤである。

1.1.5 フィールドまたはメソッドに宣言したプリミティブ型変数

フィールドまたはメソッドにプリミティブ型の変数を宣言すると、それぞれの変数に対応したレジスタが、モジュール内部に作成される。図 6 にプリミティブ型変数から作成されるレジスタ例を示す。

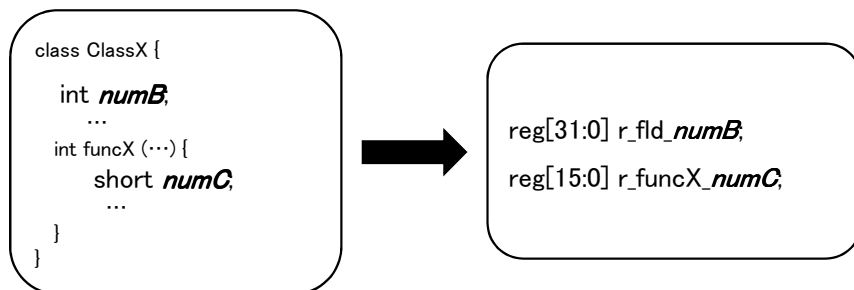


図 6: フィールドまたはメソッドのプリミティブ型変数 → レジスタ

1.1.6 フィールドに宣言したクラス型変数

フィールドにクラス型の変数を宣言し、final を付けて初期化を行うと、そのオブジェクトに対応したサブモジュールが作成される。図 7 にプリミティブ型変数から作成されるレジスタ例を示す。

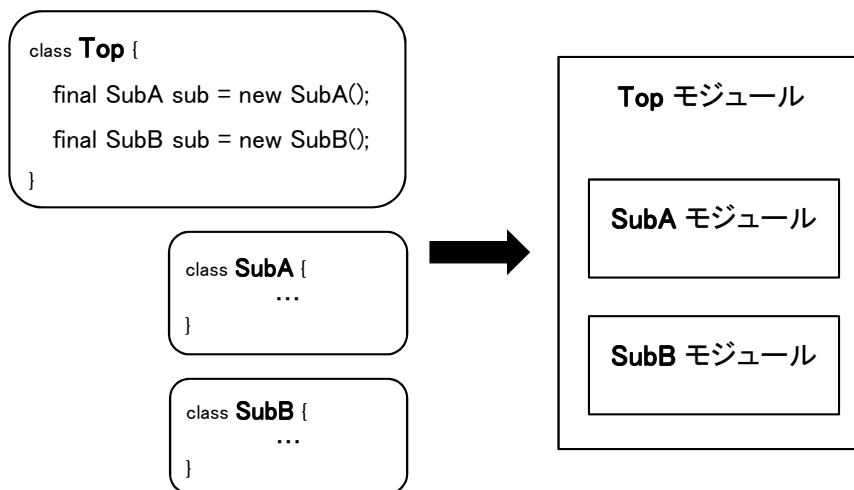


図 7: クラス型変数 → サブモジュール