コンフィグファイル説明

1 各設定項目の詳細

bitWidth

Java における各プリミティブ型変数から作成されるレジスタの bit 幅を指定. 各型は, Java で定義された bit 幅以上に設定できない. また, bit 幅は byte < short(=char) < int < long でなければならない.

byteBitWidth

byte 型変数から作成されるレジスタのビット幅.

shortBitWidth

short 型変数から作成されるレジスタのビット幅. char は short と同一の bit 幅に設定される.

intBitWidth

int 型変数から作成されるレジスタのビット幅.

longBitWidth

long 型変数から作成されるレジスタのビット幅.

floatingIsSingle

JavaRock-Thrash の浮動小数点型は, float もしくは, double のどちらか一方のみ使用可能. true の場合, float が使用可能で, false の場合 double が使用可能となる.

haveClockEnablePort

作成される Verilog モジュールに clockEnable ポートを設ける．true の場合，利用する全 IP コアに自動的に clockEnable ポートが付加される.

IPcore

Java ソースファイルに記述した演算子や配列に対し割り当てられる IP コアの設定. 各 IP コアの機能とポートの詳細については, IPcore_interface.pdf に記載.

latency

IP コアにデータを入力してから結果が出力されるまでのクロック数.

throughput

IP コアにデータを入力してから, 次に値を入力できるようになるまでのクロック数. 1 ならば, 毎クロックデータを入力可能. 2 以上の場合, haveNewDataPort を true にしなければ正しい結果が得られない.

moduleName

IP コアのモジュール名.

availableNum

モジュール内部で利用可能な, その IP コアの最大個数.

useIP

対応する演算に対し, IP コアを利用するかしないかの設定. true ならば利用する. 利用しない場合は, Verilog HDL の演算子を用いる. 浮動小数点演算は, 必ず IP コアを利用しなければならないため, この項目は存在しない.

***Pname**

IP コアのポートの名前

***BitWidth**

ポートの bit 幅. 各演算は, ここに指定した bit 幅以上の値を正しく計算できない. もし, 64bit の整数乗算を行いたいならば, multInt の multiplicandBitWidth , multiplierBitWidth , productBitWidth を 64 以上に設定しなければならない.

haveNewDataPort

newData ポート (IP コアに新しいデータを入力することを知らせる信号) を付けるかどうか. true で付ける. throughput を 2 以上に設定した場合, true にしなければ正しい結果が得られない.

generateCode

Verilog のコード生成に関する指定を行う.

positiveEdge

true でポジティブエッジで動作. false でネガティブエッジで動作.

negativeReset

true で ネガティブリセット. false でポジティブリセット.

paramListArrayAddrBitWidth

パラメータリストに記述した配列から作成される BRAM 制御用ポートのアドレス信号の bit 幅.

genCodeForCompressedState

ステート数増加による回路規模の増加を抑えるような Verilog コードを生成する.

chaining

各演算に対しチェイニングを実行するかどうかの指定. false の場合, その演算の結果が必ずレジスタに代入される.

addInt

整数型の加減算に対するチェイニング.

multInt

整数型の乗算に対するチェイニング.

divInt

整数型の除算, 剰余算に対するチェイニング.

unaryMinus

単項演算子の '−' に対するチェイニング.

bitOP

&, |, ^, ~ に対するチェイニング.

logicalOp

||, &&, ! に対するチェイニング.

compareOp

==, ! =, <, <=, >=, > に対するチェイニング.

shiftOp

シフト演算に対するチェイニング.

registerSharing

一時レジスタ（演算結果を一時的に保持しておくレジスタ）をシェアリングする最大ステート数を決定する.

maxIntSharingNum

int 型の一時レジスタをシェアリングする最大ステート数

maxFloarSharingNum

浮動小数点型の一時レジスタをシェアリングする最大ステート数

maxLongSharingNum

long 型の一時レジスタをシェアリングする最大ステート数

maxBoolSharingNum

boolean 型の一時レジスタをシェアリングする最大ステート数

scheduling

スケジューリングに関する設定を行う.

saveReadVariableWithWARhazard

true : WAR ハザードを回避するため，読み取り変数の値を一時レジスタに退避する.

false : WAR ハザードを回避するため，変数への書き込みタイミングを遅らせる.

forwardingEnable

true :

a = b+c;

x = a*a;

のように書いた場合，加算結果を a に格納すると同時に，可能であれば乗算器にも入力する.

false : a に加算結果を格納した次のステップ以降に a の値を乗算器にか入力する.

reduceConnectionConst

true : マルチプレクサが少なくなるようにバインディングを行う.

assignSameIPtoUnrolledNode

true : ループ展開をした際，コピーされた式と元の式の対応する演算子に同一の IP コアを割り当てるよう試みる．reduceConnectionConst と共に true にすることで, マルチプレクサ数が少なくなる可能性がある.