

Nios[®] II 倍精度浮動小数演算 カスタム・インストラクションの実装

Ver.17.1



Nios®II- 倍精度浮動小数演算カスタム・インストラクションの実装

1.	はじめに	4
2.	適用条件	4
	2-1. 対応バージョン	4
	2-2. 検証ハードウェア	4
3.	仕様	4
	3-1. 実装される演算	4
	3-2. 使用する IP コア	5
	3-3. ソフトウェアからの実行	5
	3-3-1. コンパイル・オプション	5
	3-4. コンパイラ・オプション設定	6
	3-5. 簡易ブロック・ダイヤグラム	6
	3-6. 四則演算の動作シーケンス	7
	3-7. 整数(32bit)から倍精度浮動小数への型変換のシーケンス	9
	3-8. 倍精度浮動小数⇒整数(32bit)への型変換のシーケンス	10
	3-9 比較演算のシーケンス	11
4.	実装	.13
4.	実装 実装 4-1. ベース・プロジェクトの準備	13 13
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成	13 13 13
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算⊐アの生成	13 13 13 13
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数乗算コアの生成	13 13 13 13 16
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数乗算コアの生成 4-2-3. 倍精度浮動小数除算コアの生成	.13 13 13 13 13 16 17
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数乗算コアの生成 4-2-3. 倍精度浮動小数除算コアの生成	.13 13 13 13 13 16 17 19
4.	 まま 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成	.13 13 13 13 13 16 17 19 20
4.	 ま、比較展昇のシークシベニニニー 実装	.13 13 13 13 16 17 19 20 22
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数乗算コアの生成 4-2-3. 倍精度浮動小数除算コアの生成 4-2-4. 整数(32bit)から倍精度浮動小数への変換コアの生成 4-2-5. 倍精度浮動小数から整数(32bit)への変換コアの生成 4-2-6. 倍精度浮動小数比較コアの生成	.13 13 13 13 16 17 19 20 22 24
4.	 実装	13 13 13 13 16 17 19 20 22 24 27
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数乗算コアの生成 4-2-3. 倍精度浮動小数除算コアの生成 4-2-4. 整数(32bit)から倍精度浮動小数への変換コアの生成 4-2-5. 倍精度浮動小数比較コアの生成 4-2-6. 倍精度浮動小数比較コアの生成 4-3. カスタム・インストラクションのトップ・レベル・ファイル 4-4. Component Editor によるカスタム・インストラクションの作成 検証	13 13 13 13 16 17 19 20 22 24 27 32
4.	 実装	13 13 13 13 16 17 19 20 22 22 24 27 32 32
4.	実装 4-1. ベース・プロジェクトの準備 4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成 4-2-1. 倍精度浮動小数加減算コアの生成 4-2-2. 倍精度浮動小数除算コアの生成 4-2-3. 倍精度浮動小数除算コアの生成 4-2-4. 整数(32bit)から倍精度浮動小数への変換コアの生成 4-2-5. 倍精度浮動小数から整数(32bit)への変換コアの生成 4-2-6. 倍精度浮動小数比較コアの生成 4-3. カスタム・インストラクションのトップ・レベル・ファイル 4-4. Component Editor 「こよるカスタム・インストラクションの作成 検証 5-1. ModelSim-Intel FPGA Edition 10.5b (以降、ModelSim)によるシミュレーション 5-1-1. 四則演算の確認	13 13 13 13 16 17 19 20 22 22 22 32 32 32



Nios®Ⅱ- 倍精度浮動小数演算カスタム・インストラクションの実装

5-1-3. 比較演算の確認	
5-2. 実機動作による性能の計測	
5-3. 使用リソース	
6. 補足	41
6-1. 注意事項	
改版履歴	42

3/42

1. <u>はじめに</u>

Nios® II にオプションで実装できる浮動小数演算カスタム・インストラクションでは、単精度浮動小数の四則演 算はサポートしているものの倍精度浮動小数演算はサポートされておらず、倍精度浮動小数演算はソフトウェア 演算で実現しています。しかし、ソフトウェアでの演算では、実行速度が非常に遅く、倍精度浮動小数演算を多用 するアプリケーションでは、プロセッサの要求性能を満たすことができない場合があります。そこで、倍精度浮動 小数演算をハードウェアで行わせることで、高い演算性能を実現できます。

本資料では、倍精度浮動小数の四則演算、型変換および比較をカスタム・インストラクションで実装する手順 およびサンプル・コードを紹介します。

2. 適用条件

2-1. 対応バージョン

本資料では、下記のツール、バージョンを使用しています。

- Intel[®] Quartus[®] Prime Version 17.1.0
- Nios[®] II Software Build Tools for Eclipse 17.1.0
- ModelSim[®]-Intel[®] FPGA Edition 10.5b
- ※ 17.1 以前のバージョンでも同様の方法で実装することは可能ですが、一部の機能や操作方法が異なる 場合がありますのでご注意ください。
- 2-2. 検証ハードウェア
 - Atlas-SoC Kit / DEO-Nano-SoC Kit

(FPGA: Cyclone® V SE 5CSEMA4U23C6N)

http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=941&PartNo=4

3. <u>仕様</u>

3-1. 実装される演算

本資料では、下記の12の倍精度浮動小数演算を実装しています。

- 四則演算(加減乗除の 4 演算)
- 浮動小数から整数(32 bit)への型変換
- 整数(32 bit)から浮動小数への型変換
- 比較演算(大小、等価、非等価の 6 比較)

これらの演算を行うための HDL コードは IP Catalog を使用して生成し、カスタム・インストラクションのトップ・ レベル・モジュールは自作しています。

3-2. 使用する IP コア

本実装で使用する IP コアは、比較演算を除いて ALTERA_FP_FUNCTION コアを使用して生成し、比較演算は ALTFP_COMPARE コアを使用して生成しています。

加減算については1つのコアを使って加算/減算モードを切り替えて動作させます。また、倍精度浮動小数と 整数の型変換については、1つのコアでは単方向しかできませんので、倍精度浮動小数から整数への変換に1 つ、整数から倍精度浮動小数への変換に1つ、それぞれ実装します。

3-3. ソフトウェアからの実行

GCC for Nios II の言語拡張機能には、これらの倍精度浮動小数演算のカスタム・インストラクションを、C 言語 の一般的なオペレータで実行するためのコンパイラ・オプションが用意されています。ただし、 Nios II は 32bit レジスタ・セットしか持っていませんので、 64bit の倍精度浮動小数を直接カスタム・インストラクション・モジュー ルに入出力することはできません。そのため、言語拡張機能による実装を行う場合には、複数のステートを持つ カスタム・インストラクションを決められたシーケンスで実行できるように実装する必要があります。

なお、上記、言語拡張機能を使用せず、カスタム・インストラクションを直接使用して演算を行う場合は、決めら れたシーケンスに従う必要はありません。

3-3-1. コンパイル・オプション

-mcustom-fwrx = 0	・・・・ 64bit を内部レジスタに書き込む
-mcustom-fwry = 1	・・・・ 32bit を内部レジスタに書き込む
-mcustom-frdxlo = 2	・・・・ 内部レジスタの下位 32bit を読み出す
-mcustom-frdxhi = 3	・・・・ 内部レジスタの上位 32bit を読み出す
-mcustom-frdy = 4	・・・・ 内部レジスタの下位 32bit を読み出す
-mcustom-faddd = 5	・・・・ 加算を実行
-mcustom-fsubd = 6	・・・・ 減算を実行
-mcustom-fmuld = 7	・・・・ 乗算を実行
-mcustom-fdivd = 8	・・・・ 除算を実行
-mcustom-floatid = 9	・・・・ 整数を倍精度浮動小数に変換
-mcustom-fixdi = 10	・・・・ 倍精度浮動小数を整数に変換
-mcustom-fcmpltd = 11	・・・・ < 比較を実行
-mcustom-fcmpled = 12	・・・・ <= 比較を実行
-mcustom-fcmpgtd = 13	・・・・ > 比較を実行
-mcustom-fcmpged = 14	・・・・ >= 比較を実行
-mcustom-fcmpeqd = 15	・・・・ == 比較を実行
-mcustom-fcmpned = 16	・・・・ != 比較を実行

※ "="の後ろの数字は、カスタム・インストラクションの相対番号を示しています。



3-4. コンパイラ・オプション設定

コンパイル・オプションの設定は、Makefile を直接編集してください。アプリケーション・プロジェクトのフォル ダにある "Makefile" をテキスト・エディタで開き、"APP_CFLAGS_USER_FLAGS :=" にコンパイラ・オプションを記述 してください。



APP_CFLAGS_USER_FLAGS := -mcustom-fwrx=224 -mcustom-fwry=225 -mcustom-frdxlo=226 -mcustom-frdxhi=227 -mcustom-frdy=228 -mcustom-faddd=229 -mcustom-fsubd=230 -mcustom-fmuld=231 -mcustom-fdivd=232 -mcustom-floatid=233 -mcustom-fixdi=234 -mcustom-fcmpltd=235 -mcustom-fcmpled=236 -mcustom-fcmpgtd=237 -mcustom-fcmpged=238 -mcustom-fcmpeqd=239 -mcustom-fcmpned=240

※上記のカスタム・インストラクションの番号は、Platform Designer にて開始オペコードを"224"に設定した場合です。他の値に設定した場合は、開始オペコード+相対番号で設定してください。



3-5. 簡易ブロック・ダイヤグラム

3-6. 四則演算の動作シーケンス

1. 入力 A の 64bit データを、2 つのレジスタを使ってカスタム・インストラクション・モジュールに入力し、モジ ュール内部のレジスタに 64bit データを保持します。



2. 入力 B の 64bit データを、2 つのレジスタを使ってカスタム・インストラクション・モジュールに入力し、演算 完了を待ち、演算結果の上位 32bit を出力します。



3. カスタム・インストラクション・モジュールから演算結果の下位 32bit を出力して演算結果 64bit を合成しま す。



{

【C 言語記述とアセンブラ・コードの例】

Γ

double	A = 1.23456789;
double	B = 1.23456789;
volatile	double R;
R = A + B	3;

R = A + B; R = A * B; R = A / B;

29c: 0090a134 movhi r2,17028 2a0: 10b786c4 addi r2,r2,-8677 2a4: e0bffa15 stw r2,-24(fp) 2a8: 0098fd34 movhi r2,16372 2ac: 10b03284 addi r2,r2,-16182 2b0: e0bffb15 stw r2,-20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-16182 2c8: e0bffd15 stw r2,-16182 2c8: e0bffd15 stw r2,-1617 2d0: e17ffb17 ldw r3,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d2: 2141f832 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-16(fp) 2d4: e0fff17 ldw r3,-12(fp) 2d4: e005f917 ldw r3,-12(fp) 2d4: e005f917 ldw r3,-12(fp) 2d5: 2141f832 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r3,-4(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f4: e17ffb17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 2e8: e0bffe15 stw r3,-4(fp) 2f4: e17ffb17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 228,r2,zero,zero 305: e0bffe15 stw r3,-4(fp) 310: e0fff15 stw r3,-4(fp) 312: e0bffc17 ldw r4,-24(fp) 313: e17ffb17 ldw r4,-24(fp) 314: e13ffa17 ldw r4,-24(fp) 315: e0bffc17 ldw r3,-12(fp) 326: 0005f932 custom 228,r2,zero,zero 307: e0bffe15 stw r3,-4(fp) 328: e13ffa17 ldw r4,-24(fp) 314: e13ffa17 ldw r3,-12(fp) 326: 0005f932 custom 228,r2,zero,zero 307: e0bffe15 stw r3,-4(fp) 328: e10c7f9f2 custom 228,r2,zero,zero 330: e0bffe15 stw r3,-4(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 345: e0bffc17 ldw r3,-12(fp) 346: e0fff17 ldw r3,-12(fp) 347: e0fff15 stw r3,-4(fp) 348: e13ffa17 ldw r4,-24(fp) 336: e0fff15 stw r3,-4(fp) 341: e0fff17 ldw r3,-12(fp) 342: 2141f832 custom 228,r2,zero,zero 353: e005f932 custom 228,r2,zero,zero 354: e00ffe15 stw r3,-4(fp)	Ľ	double	A = 1.234567	/89;		
2a0: 10b786c4 addi r2,r2,-8677 2a4: e0bffa15 stw r2,-24(fp) 2a8: 008ffd34 movhi r2,16372 2ac: 10b03284 addi r2,r2,-16182 2b0: e0bffb15 stw r2,-20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) 2d0: e17ffb17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d2: e17ffb17 ldw r3,-12(fp) 2d3: e0fffd17 ldw r3,-12(fp) 2d4: e0bffc15 stw r2,-8(fp) 2d6: e10ff15 stw r2,-8(fp) 2d6: e13ffa17 ldw r4,-24(fp) 2d7: e13ffa17 ldw r4,-24(fp) 2d8: e0fff15 stw r3,-4(fp) R = A + B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f4: e13ffa17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 2f8: e0bffe15 stw r3,-4(fp) R = A - B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,r4,r5 304: 10c7f9b2 custom 228,r2,zero,zero 305: e0bffe15 stw r2,-8(fp) 310: e0fff17 ldw r3,-12(fp) 310: e0fff17 ldw r3,-2(fp) 311: e13ffa17 ldw r4,-24(fp) 312: e13ffa17 ldw r4,-24(fp) 313: e17ffb17 ldw r5,-20(fp) 314: e13ffa17 ldw r4,-24(fp) 315: e0bffc17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 305: e0bffe15 stw r2,-8(fp) 316: e0bffc17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A + B; 344: e13ffa17 ldw r4,-24(fp) 335: e13ffa17 ldw r4,-24(fp) 336: e0bffe15 stw r2,-8(fp) 337; r2,r3 327; 0005f932 custom 228,r2,zero,zero 330: e0bffe15 stw r3,-4(fp) R = A / B; 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r4,-24(fp) 345: e0ff17 ldw r4,-24(fp) 346: e0fff17 ldw r4,-24(fp) 347; e0fff15 stw r3,-4(fp) 348: e0fff15 stw r3,-4(fp) 349: e0fff15 stw r3,-4(fp) 340: e0bffc15 s		29c:	0090a134	movhi	r2,170)28
2a4: e0bffa15 stw r2,-24(fp) 2a8: 008ffd34 movhi r2,16372 2ac: 10b03284 addi r2,r2,-16182 2b0: e0bffb15 stw r2,-20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-3677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; R = A + B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d2: e141f832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2ec: e0fff17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-20(fp) 2f4: e10ffb17 ldw r3,-20(fp) 2f5: e00ffc17 ldw r3,-20(fp) 2f6: e00fff17 ldw r3,-20(fp) 2f6: e00fff15 stw r2,-8(fp) 300: 2141f832 custom 228,r2,zero,zero 300: e00fff15 stw r3,-4(fp) R = A + B; 314: e13ffa17 ldw r4,-24(fp) 310: e00fff15 stw r3,-4(fp) 312: e00fff17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 305: e00fff17 ldw r3,-12(fp) 310: e00fff17 ldw r3,-12(fp) 322: e00ff17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 305: e00ff615 stw r2,-8(fp) 316: e00ff617 ldw r3,-12(fp) 326: 0005f932 custom 228,r2,zero,zero 307: e00ff615 stw r2,-8(fp) 318: e17ffb17 ldw r3,-20(fp) 324: 2141f832 custom 228,r2,zero,zero 330: e00ffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) 344: e0fff17 ldw r4,-24(fp) 335: e13ffa17 ldw r4,-24(fp) 336: e107f92 custom 228,r2,zero,zero 330: e00ffe15 stw r3,-4(fp) 344: e0fff17 ldw r5,-20(fp) 344: e0fff17 ldw r4,-24(fp) 355: 0005f932 custom 228,r2,zero,zero 360: e00ffe15 stw r3,-4(fp) 360: 0005f932 custom 228,r2,zero,zero 361: e00ffe15 stw r3,-4(fp) 362: e00ffe15 stw r3,-4(fp) 363: e00ffe15 stw r3,-4(fp) 364: e00ffe15 stw r3,-4(fp) 365: 0005f932 custom 228,r2,zero,zero		2a0:	10b786c4	addi	r2.r28	8677
288: 008ffd34 movhi r2,16377 2ac: 10b03284 addi r2,r2,-16182 2b0: e0bffb15 stw r2,-20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; R = A + B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d6: 2141f832 custom 228,r2,zero,r4,r5 2e0: 10c7f972 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2dc: e13ffa17 ldw r4,-24(fp) 2f6: e13ffa17 ldw r3,-12(fp) 7f6: e13ffa17 ldw r3,-12(fp) 2f6: e0fff17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) 311: e0bffc17 ldw r4,-24(fp) 312: e0bffc17 ldw r4,-24(fp) 313: e17ffb17 ldw r5,-20(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 305: e0bffe15 stw r2,-8(fp) 326: e00ffe15 stw r2,-8(fp) 327; 2141f832 custom 228,r2,zero,r4,r5 328: 10c7f92 custom 231,r3,r2,r3 320: 0005f932 custom 228,r2,zero,r4,r5 328: 10c7f97 custom 224,r2,r3 320: 0005f932 custom 224,r2,r3 350: 0005f932 custom 224,r2,r6,r5 344: e0ffd17 ldw r3,-12(fp) 344: e0ffd17 ldw r3,-12(fp) 345: e0fff17 ldw r3,-12(fp) 346: 204ff15 stw r2,-8(fp) 358: e0fff15 stw r		2a4:	e0bffa15	stw	r224	(fp)
2ac: 10b03284 addi r2,r2,-16182 2b0: e0bffb15 stw r2,r20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b7864 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,17028 2c8: e0bffd15 stw r2,-12(fp) volatile double R; R=A+B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d6: e17ffb17 ldw r3,-12(fp) 2d6: e17ffb17 ldw r4,-24(fp) 2d7: 2141f832 custom 224,zero,r4,r5 2e0: 1007f972 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2c1: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f5: e0fffd17 ldw r3,-12(fp) 2f6: e0fffd17 ldw r3,-12(fp) 2f6: e0fffd17 ldw r3,-12(fp) 2f6: e0fffd17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 1007f92 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R=A+B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e0fff15 stw r2,-8(fp) 310: e0fff15 stw r2,-8(fp) 310: e0fff15 stw r2,-8(fp) 310: e0fff17 ldw r3,-12(fp) 324: 2141f832 custom 224,zero,r4,r5 328: 1007f9f2 custom 224,zero,r4,r5 328: 2141f832 custom 224,zero,r4,r5 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 345: e0fff17 ldw r3,-12(fp) 346: 2141f832 custom 224,zero,zero 356: 0005f932 custom 228,r2,zero,zero 357: 0005f932 custom 228,r2,zero,zero 358: e0fff15 stw r3,-4(fp)		2a8:	008ffd34	movhi	r2.163	572
2b0: e00ffb15 stw r2,-20(fp) double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e00ffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e00ffd15 stw r2,-12(fp) volatile double R; R = A + B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e00ffc17 ldw r3,-12(fp) 2d2: 2141f832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 228,r2,zero,zero 2e8: e00ffe15 stw r2,-4(fp) 2d4: e17ffb17 ldw r4,-24(fp) 2d2: e0fff17 ldw r3,-12(fp) 2d2: 2141f832 custom 228,r2,zero,zero 2e8: e00ffe15 stw r3,-4(fp) R = A - B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f952 custom 228,r2,zero,zero 30c: e00ffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 316: e00ffc17 ldw r5,-20(fp) 316: e00ffc17 ldw r3,-12(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e0fff15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e0fff15 stw r2,-8(fp) 320: e0fff17 ldw r3,-12(fp) 324: 2141f832 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f912 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f932 custom 224,zero,r4,r5 328: 10c7f932 custom 224,zero,r4,r5 330: e0bffe15 stw r2,-8(fp) 334: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 345: e0fff15 stw r2,-8(fp) 356: 0005f932 custom 228,r2,zero,zero 354: e00ff15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		2ac:	10b03284	addi	r2.r2	16182
double B = 1.23456789; 2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-16(fp) 2d0: e17ffb17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r2,-16(fp) 2d2: 2141832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2ec: e0fff17 ldw r4,-24(fp) 2d2: e0fff17 ldw r4,-24(fp) 2d2: e0fff15 stw r2,-8(fp) 2ec: e0fff17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f5: e00ffc15 stw r2,-8(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 223,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e00ffe15 stw r2,-8(fp) 310: e00ffc15 stw r3,-4(fp) R = A + B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e00ffc17 ldw r3,-12(fp) 310: e00ffc17 ldw r3,-12(fp) 310: e00ffc17 ldw r3,-12(fp) 310: e00ffc17 ldw r3,-12(fp) 320: e00ffe15 stw r2,-8(fp) 310: e00ffc17 ldw r3,-12(fp) 320: e00ffc17 ldw r3,-12(fp) 320: e00ffc17 ldw r3,-12(fp) 320: e00ffc17 ldw r3,-12(fp) 320: e00ffc15 stw r2,-8(fp) 334: e01ff17 ldw r3,-12(fp) 344: e13ffa17 ldw r4,-24(fp) 334: e01ff17 ldw r3,-12(fp) 344: e01ff17 ldw r3,-12(fp) 345: e01ff15 stw r2,-8(fp) 350: 0005f932 custom 223,r3,r2,r3 350: 0005f932		2b0:	e0bffb15	stw	r220	(fp)
2b4: 0090a134 movhi r2,17028 2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; $\mathbf{R} = \mathbf{A} + \mathbf{B};$ 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d5: 2141832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2d6: e13ffa17 ldw r4,-24(fp) 2d7: e13ffa17 ldw r3,-12(fp) 2d8: e0fff11 ldw r3,-12(fp) 2d8: e0fff11 ldw r3,-12(fp) 2d8: e0fff15 stw r2,-8(fp) 2d9: e13ffa17 ldw r3,-12(fp) 2f8: e0bffc17 ldw r3,-12(fp) 2f8: e0bffc17 ldw r3,-12(fp) 300: 2141832 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 229,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff17 ldw r3,-12(fp) 320: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r3,-12(fp) 322: e005f932 custom 228,r2,zero,zero 303: e0bffe15 stw r2,-8(fp) 314: e13ffa17 ldw r3,-12(fp) 325: e17ffb17 ldw r3,-12(fp) 326: 0005f932 custom 224,zero,r4,r5 328: 10c7f92 custom 231,r3,r2,r3 320: 0005f932 custom 224,zero,r4,r5 328: 10c7f92 custom 224,zero,r4,r5 328: 10c7f912 custom 224,zero,r4,r5 328: e13ffa17 ldw r4,-24(fp) 334: e0fff115 stw r2,-8(fp) 334: e0fff115 stw r2,-8(fp) 334: e0fff117 ldw r5,-20(fp) 344: e0fff117 ldw r5,-20(fp) 344: e0fff117 ldw r5,-20(fp) 344: e0fff117 ldw r3,-12(fp) 344: e0fff15 stw r2,-8(fp) 345: e13ff317 ldw r4,-24(fp) 346: 2141832 custom 224,zero,r4,r5 347: e0bffc15 stw r2,-8(fp) 348: 2141832 custom 224,zero,r4,r5 347: e0bffc15 stw r2,-8(fp) 348: 2141832 custom 224,zero,r4,r5 347: e0bffc15 stw r2,-8		double	B = 1.234567	89.	12, 20,	('P')
2b8: 10b786c4 addi r2,r2,-8677 2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; $\mathbf{R} = \mathbf{A} + \mathbf{B};$ 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2dc: 2141f832 custom 229,r3,r2,r3 2e4: 0005f932 custom 229,r3,r2,r3 2e4: 0005f932 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r3,-4(fp) $\mathbf{R} = \mathbf{A} - \mathbf{B};$ 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f4: e0bffc17 ldw r3,-12(fp) 2f6: e0fffd17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 300: 2141f832 custom 228,r2,zero,zero 300: e0bffe15 stw r2,-8(fp) 310: e0fff17 ldw r5,-20(fp) 310: e0fff17 ldw r5,-20(fp) 310: e0fff17 ldw r5,-20(fp) 310: e0fff17 ldw r4,-24(fp) 311: e0bffc17 ldw r5,-20(fp) 322: e0bffe15 stw r2,-8(fp) 312: e0bffe15 stw r2,-8(fp) 313: e17ffb17 ldw r5,-20(fp) 324: 2141f832 custom 228,r2,zero,zero 305: e0bffe15 stw r2,-8(fp) 316: e0bffe15 stw r2,-8(fp) 317: e0bffe17 ldw r5,-20(fp) 328: 10c7f92 custom 221,r3,r2,r3 329: 0005f932 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff17 ldw r5,-20(fp) 324: 2141f832 custom 224,zero,r4,r5 328: 10c7f92 custom 231,r3,r2,r3 320: 0005f932 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A/B; 338: e13ffa17 ldw r4,-24(fp) 334: e0fff15 stw r2,-8(fp) 334: e0fff15 stw r2,-8(fp) 334: e0fff15 stw r2,-8(fp) 334: e0fff15 stw r2,-8(fp) 335: e13ffa17 ldw r4,-24(fp) 336: e13ffa17 ldw r4,-24(fp) 337: e17fb17 ldw r5,-20(fp) 348: 2141f832 custom 224,zero,r4,r5 340: e0bffc17 ldw r3,-2(fp) 344: e0fff17 ldw r3,-2(fp) 345: e13ffa17 stw r3,-4(fp) 346: 2141f832 custom 232,r3,r2,r3 350: 0005f932		2h4·	0090a134	movhi	r2 170	128
2bc: e0bffc15 stw r2,-16(fp) 2c0: 008ffd34 movhi r2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; R=A+B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r3,-12(fp) 2d8: e0fffd17 ldw r3,-12(fp) 2d8: e0fffd15 stw r2,-8(fp) 2d9: e0fff15 stw r2,-8(fp) 2e1: e0fff17 ldw r3,-12(fp) 2f3: e0bffc17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f3: e0bffc17 ldw r3,-12(fp) 2f3: e0bffc17 ldw r3,-12(fp) 2f4: e17ffb17 ldw r3,-12(fp) 2f3: e0bffc17 ldw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 300: e0bffe15 stw r3,-4(fp) 7f3: e0bffc17 ldw r4,-24(fp) 310: e0fff15 stw r3,-4(fp) 310: e0fff15 stw r3,-12(fp) 310: e0fff15 stw r3,-2(fp) 310: e0fff17 ldw r4,-24(fp) 311: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r4,-24(fp) 313: e17fb17 ldw r3,-12(fp) 324: 2141f832 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 328: 0005f932 custom 231,r3,r2,r3 328: 0005f932 custom 231,r3,r2,r3 328: 0005f932 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 329: 0005f932 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 320: e0bffc15 stw r3,-4(fp) R=A/B; 338: e13ffa17 ldw r4,-24(fp) 334: e0fff15 stw r3,-4(fp) 344: e0fff17 ldw r5,-20(fp) 344: e0fff17 ldw r5,-20(fp) 344: e0fff17 ldw r4,-24(fp) 345: e13ffa17 ldw r4,-24(fp) 346: 2141f832 custom 224,zero,r4,r5 326: 0005f932 custom 224,zero,r4,r5 327,3,r2,r3 350: 0005f932 custom 232,r3,r2,r3 350: 0005f932 custo		2b8:	10b786c4	addi	r2.r2	8677
2c0: 008ffd34 movhir2,16372 2c4: 10b03284 addi r2,r2,-16182 2c8: e0bffd15 stw r2,-12(fp) volatile double R; R=A+B; 2cc: e13ffa17 ldw r4,-24(fp) 2d0: e17ffb17 ldw r5,-20(fp) 2d4: e0bffc17 ldw r2,-16(fp) 2d8: e0fffd17 ldw r3,-12(fp) 2d6: 2141f832 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2ec: e0fff15 stw r3,-4(fp) R=A-B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e13ffa17 ldw r3,-12(fp) 2f4: e13ffa17 ldw r3,-12(fp) 2f5: e0fff15 stw r3,-4(fp) R=A+B; 2f0: e13ffa17 ldw r4,-24(fp) 2f6: e0fff17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f922 custom 224,zero,r4,r5 304: 10c7f92 custom 224,zero,r4,r5 304: 10c7f92 custom 224,zero,r4,r5 305: 204ff15 stw r3,-4(fp) R=A+B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff15 stw r3,-4(fp) R=A+B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff15 stw r3,-4(fp) R=A+B; 314: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 321: e0bffc17 ldw r3,-4(fp) R=A/B; 338: e13ffa17 ldw r4,-24(fp) 339: e0bffe15 stw r2,-8(fp) 330: e0bffe15 stw r2,-8(fp) 331: e13ffa17 ldw r4,-24(fp) 332: e13ffa17 ldw r4,-24(fp) 333: e13ffa17 ldw r4,-24(fp) 334: e0fff15 stw r2,-8(fp) 335: e13ffa17 ldw r4,-24(fp) 336: e13ffa17 ldw r4,-24(fp) 337: e13ffa17 ldw r4,-24(fp) 338: e13ffa17 ldw r4,-24(fp) 339: e0bffe15 stw r2,-8(fp) 340: e0bffc17 ldw r3,-12(fp) 341: e0fff17 ldw r3,-12(fp) 342: 2141f832 custom 224,zero,r4,r5 342: 10c7f32 custom 224,zero,r4,r5 343: e0fff15 stw r2,-8(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 345: e0fff15 stw r2,-8(fp) 346: 2141f832 custom 224,zero,r4,r5 340: 10c7f32 custom 232,r3,r2,r3 350: 000		2bc:	e0bffc15	stw	r216	(fp)
2cd:10003284 addir2,r2,r16182 r2,r2,r161822c8:e0bffd15 stwr2,r2,r16182 stw2c8:e0bffd15 stwr2,r2,r16182 stw2c6:e13ffa17 e1dwldwr4,r24(fp) r2,r16(fp) 2d0:2d0:e17ffb17 e1dwldwr5,r20(fp) r2,r16(fp) 2d3:2d1:e0bffc17 e1dwldwr3,r12(fp) r2,r16(fp)2d2:2141832 e10c7f972 customcustom224,zero,r4,r5 r2,r3 custom2e0:10c7f972 e2e3:custom228,r2,zero,zero r2,r3,r2,r3 custom2e4:0005f932 e2e3:custom224,zero,r4,r5 r2,r6(fp)2f2:e0fff15 e13ffa17 e1dwr5,-20(fp) r3,r2,r3 custom224,zero,r4,r5 custom20:2141f832 e10c7f9b2 customcustom228,r2,zero,zero r2,r2,r3 custom30:2005f932 e1171 e1dwr3,-4(fp) r3,-12(fp)30:e0bffe15 e117 e1dwr4,-24(fp) r3,-4(fp)318:e17ff171 e1dwr4,-24(fp) r3,-12(fp)321:e0bffe17 e117 e1dwcustom224,zero,r4,r5 customcustom224,zero,r4,r5 custom328:e0fff17 e117 e1dwr5,-20(fp) r3,-12(fp)324:e0fff17 e117 e1dwr4,-24(fp) r3,-12(fp)324:e0fff15 e117 e1dwr2,-8(fp) r2,-28(fp)330:e0bffe15 e0bffe15 stwr2,-8(fp) r2,-28(fp)3314:e0fff17 e1dwr4,-24(fp) r2,-28(fp)332:e0bffe		200	008ffd34	movhi	r2 163	172
2c8:e005fd15stw $r_2, -12(fp)$ volatile doubleR;R=A+B;2cc:e13ffa17ldwr4, -24(fp)2d0:e17ffb17ldwr5, -20(fp)2d4:e0bffc17ldwr3, -12(fp)2d5:e0fff17ldwr3, -12(fp)2d6:2141832custom224, zero, r4, r52e0:10c7f972custom228, r2, zero, zero2e8:e0bffe15stwr2, -8(fp)2ec:e0fff17ldwr4, -24(fp)2f3:e13ffa17ldwr4, -24(fp)2f4:e13ffa17ldwr5, -20(fp)2f6:e0fff17ldwr3, -12(fp)300:2141f832custom224, zero, r4, r5304:10c7f912custom223, r3, r2, r3308:0005f932custom224, zero, r4, r5304:10c7f912custom224, zero, r4, r5305:o005f932custom224, zero, r4, r5326:10c7f912custom224, zero, r4, r5326:10c7f912custom224, zero, r4, r5328:10c7f912custom224, zero, r4, r5328:10c7f912custom224, zero, r4, r5328:10c7f912custom224, zero, r4, r5328:10c7f912custom224, zero, r4, r5329:10		2c4	10b03284	addi	r2 r2 -	16182
volatile double R; R = A + B; 2cc: e13ffa17 dw r4,-24(fp) 2d0: e17ffb17 dw r5,-20(fp) 2d4: e0bffc17 dw r5,-20(fp) 2d4: e0bffc17 dw r5,-20(fp) 2d6: 2141f832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2ec: e0fff17 dw r5,-20(fp) 2f4: e17ffb17 dw r5,-20(fp) 2f4: e17ffb17 dw r5,-20(fp) 2f4: e00ffc17 dw r3,-12(fp) 300: 2141f832 custom 228,r2,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A + B; 314: e13ffa17 dw r4,-24(fp) 312: e0ffc17 dw r3,-12(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 dw r4,-24(fp) 312: e00ffc17 dw r3,-12(fp) 322: e00ffc17 dw r3,-12(fp) 324: 2141f832 custom 231,r3,r2,r3 328: 10c7f9f2 custom 231,r3,r2,r3 328: 10c7f9f2 custom 231,r3,r2,r3 326: 0005f932 custom 232,r3,r2,r3 326: 0005f932 custom 232,r3,r2,r3 350: 0005f9		208:	e0bffd15	stw	r2120	(fp)
R = A + B;2cc:e13ffa17ldwr4,-24(fp)2d0:e17ffb17ldwr5,-20(fp)2d4:e0bffc17ldwr3,-12(fp)2d5:c141f832custom224,zero,r4,r52e0:10C7f972custom224,zero,r4,r52e0:10C7f972custom228,r2,zero,zero2e8:e0bffe15stwr2,-8(fp)2ec:e0fff15stwr3,-4(fp)R = A - B;2f0:e13ffa17ldw2f1:e0bffc17ldwr5,-20(fp)2f2:e0fff17ldwr5,-20(fp)2f6:e0fff17ldwr5,-20(fp)2f6:e0fff17ldwr3,-12(fp)300:2141f832custom228,r2,zero,zero304:10C7f9b2custom230,r3,r2,r3308:0005f932custom228,r2,zero,zero30c:e0bffe15stwr2,-8(fp)310:e0fff15stwr3,-4(fp)R = A * B;14:e13ffa17314:e13ffa17ldwr4,-24(fp)326:e0bffc17ldwr4,-24(fp)326:e0bffe15stwr2,-8(fp)326:0005f932custom234,r3,r2,r3326:0005f932custom224,zero,r4,r5328:10C7f9f2custom231,r3,r2,r3320:e0bffe15stwr3,-2(fp)334:e0fff15stwr3,-2(fp)335:e13ffa17ldwr4,-24(fp)336:		volatile	double R:		,,	(.6)
R = A + B;2cc: $e13ffa17$ $ dw r4, -24(fp) $ 2d0: $e17ffb17$ $ dw r5, -20(fp) $ 2d4: $e0bffc17$ $ dw r3, -12(fp) $ 2d8: $e0ffd17$ $ dw r3, -12(fp) $ 2d1: $2141f832$ $custom$ $224, 2ero, r4, r5 $ 2e0: $10c7f972$ $custom$ $229, r3, r2, r3 $ 2e4: $0005f932$ $custom$ $228, r2, zero, zero $ 2e8: $e0bffe15$ $stw r2, -8(fp) $ 2e1: $e13ffa17$ $ dw r4, -24(fp) $ 2f4: $e17ffb17$ $ dw r5, -20(fp) $ 2f6: $e0fff17$ $ dw r3, -12(fp) $ 2f6: $e0fff17$ $ dw r3, -12(fp) $ 300: $2141f832$ $custom$ 230, r3, r2, r3 308 $0005f932$ 306: $0005f932$ $custom$ 307: $e0bffe15$ $stw r2, -8(fp) $ 310: $e0fff15$ $stw r3, -4(fp) $ 311: $e13ffa17$ $ dw r4, -24(fp) $ 312: $e0bffe17$ $ dw r4, -24(fp) $ 313: $e13ffa17$ $ dw r4, -24(fp) $ 314: $e13ffa17$ $ dw r4, -24(fp) $ 321: $e00ffe17$ $ dw r3, -12(fp) $ 324: $2141f832$ $custom$ 234, r2, r2, r3 $224, 2ero, r4, r5 $ 328: $10c7f9f2$ $custom$ 329: $e00ffe15$ $stw r2, -8(fp) $ 330: $e00ffe15$ $stw r3, -4(fp) $ 331: $e13ffa17 $						
2cc:e13fta17Idwr4,-24(tp)2d0:e17ffb17Idwr5,-20(tp)2d4:e0bffc17Idwr5,-20(tp)2d8:e0fffd17Idwr3,-12(tp)2dc:2141f832custom224,zero,r4,r52e0:10c7f972custom229,r3,r2,r32e4:0005f932custom228,r2,zero,zero2e8:e0bffe15stwr2,-8(tp)2ec:e0fff15stwr3,-4(tp)R=A-B;2f0:e13ffa17Idw2f0:e13ffa17Idwr4,-24(tp)2f4:e17ffb17Idwr5,-20(tp)2f8:e0bffc17Idwr5,-20(tp)2f8:e0bffc17Idwr5,-20(tp)2f6:e0fff17Idwr3,-12(tp)300:2141f832custom228,r2,zero,zero304:10c7f9b2custom230,r3,r2,r3308:0005f932custom228,r2,zero,zero301:e0fff15stwr3,-4(tp)318:e13ffa17Idwr4,-24(tp)318:e13ffa17Idwr3,-12(tp)324:2141f832custom234,zero,r4,r5328:10c7f9f2custom234,zero,r4,r5328:10c7f9f2custom234,zero,r4,r5328:10c7f9f2custom234,zero,r4,r5326:0005f932custom228,r2,zero,zero330:e0bffe15stwr3,-4(tp)34:e0fff15stwr3,-12(tp)3		R = A + E	3;			(6.)
200: $e1/Hb1/$ Idw $r5,-20(tp)$ 2d4: $e0bffc17$ Idw $r2,-16(fp)$ 2d5: $e0ffd17$ Idw $r3,-12(fp)$ 2dc:2141f832 $custom$ 224,zero,r4,r5 2e0:10C7f972 $custom$ 229,r3,r2,r3 2e4:0005f932 $custom$ 228,r2,zero,zero 2e8: $e0bffe15$ stw $r2,-8(fp)$ 2ec: $e0fff15$ stw $r3,-4(fp)$ $\mathbf{R} = \mathbf{A} - \mathbf{B};$ 2f0: $e13ffa17$ Idw2f0: $e13ffa17$ Idw $r4,-24(fp)$ 2f4: $e17ffb17$ Idw $r3,-12(fp)$ 200:2141f832 $custom$ 223,r2,r3 300:2141f832 $custom$ 223,r2,r3 301: $e0fff15$ stw $r2,-8(fp)$ 310: $e0fff15$ stw $r3,-12(fp)$ 310: $e0fff15$ stw $r3,-12(fp)$ 310: $e0fff17$ Idw $r4,-24(fp)$ 318: $e17ffb17$ Idw $r3,-12(fp)$ 324: $e0fff17$ Idw $r3,-12(fp)$ 324: $e0fff15$ stw $r2,-8(fp)$ 326: $100C792$ $custom$ 224,zero,r4,r5 328: $100C7952$ $custom$ 224,zero,r4,r5 326: $100C7952$ $custom$ 224,zero,r4,r5 326: $100C7952$ $custom$ 224,zero,r4,r5 326: $100C7952$ $custom$ 224,zero,r4,r5 330: $e0bffe15$ stw $r3,-4(fp)$ R = A / B; 338: </td <td></td> <td>2cc:</td> <td>e13ffa17</td> <td>Idw</td> <td>r4,-24(</td> <td>(TD)</td>		2cc:	e13ffa17	Idw	r4,-24((TD)
204:e0bftc17Idw $r2,-16(tp)$ 2d8:e0ffd17Idw $r3,-12(tp)$ 2dc:2141f832custom224,zero,r4,r52e0:10c7f972custom229,r3,r2,r32e4:0005f932custom228,r2,zero,zero2e8:e0bffe15stw $r2,-8(tp)$ 2ec:e0fff15stw $r3,-4(tp)$ R=A-B;Idw $r4,-24(tp)$ 2f4:e17ffb17Idw2f4:e17ffb17Idw2f5:e0fff17Idw2f6:e0fff17Idw2f6:e0fff17Idw300:2141f832custom230,r3,r2,r3custom228,r2,zero,zero304:10c7f9b2custom305:e0bffe15stw310:e0fff15stw314:e13ffa17Idwr5,-20(fp)316:e0fff17318:e17ffb17320:e0fff17321:0005f932224,zero,r4,r5328:10c7f9f2320:e0fff15321:0005f932224,zero,r4,r5322:0005f932333:e0fff15344:e0fff15351:e0fff17362:107f9f2374:e0fff15374:g1fa17375:g2, e17ffb17376:g2, e2, e2, e2, e2, e2, e2, e2, e2, e2, e		2d0:	e17ttb17	ldw	r5,-20((tp)
288:e0fftd17Idw $r_3,-12(tp)$ 2dc:2141f832custom224,zero,r4,r52e0:10c7f972custom228,r2,zero,zero2e8:e0bffe15stw $r_2,-8(tp)$ 2ec:e0fff15stw $r_3,-4(tp)$ R=A-B;Idwr4,-24(tp)2f4:e17ffb17Idwr5,-20(tp)2f6:e0fff17Idwr3,-12(tp)300:2141f832custom224,zero,r4,r5304:10c7f9b2custom230,r3,r2,r3308:0005f932custom228,r2,zero,zero30c:e0bffe15stwr2,-8(tp)310:e0fff15stwr3,-4(tp)314:e13ffa17Idwr4,-24(fp)316:e0fff17Idwr4,-24(fp)317:e0fff17Idwr4,-24(fp)318:e17ffb17Idwr5,-20(fp)310:e0fff17Idwr5,-12(tp)321:e0fff17Idwr3,-12(tp)322:0005f932custom224,zero,r4,r5328:10c7f9f2custom231,r3,r2,r3320:e0fff17Idwr4,-24(tp)331:e13ffa17Idwr4,-24(tp)322:0005f932custom224,zero,r4,r5328:10c7f9f2custom231,r3,r2,r3320:e0bffe15stwr2,-8(tp)331:e13ffa17Idwr4,-24(tp)332:e13ffa17Idwr4,-24(tp)332:005f932 <t< td=""><td></td><td>2d4:</td><td>e0bttc17</td><td>ldw</td><td>r2,-16</td><td>(tp)</td></t<>		2d4:	e0bttc17	ldw	r2,-16	(tp)
2dc: 2141f832 custom 224,zero,r4,r5 2e0: 10c7f972 custom 229,r3,r2,r3 2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r3,-4(fp) R = A - B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f8: e0bffc17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 226,r2,r3 308: 0005f932 custom 226,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff15 stw r3,-12(fp) 320: e0fff17 ldw r5,-20(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r5,-20(fp) 320: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 324: 2141f832 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 332: e17ffb17 ldw r5,-20(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-12(fp) 344: e0fff17 ldw r3,-20(fp) 340: e0bffe15 stw r2,-8(fp) 350: 0005f932 custom 228,r2,zero,zero 361: e0bffe17 ldw r3,-20(fp) 362: e17ffb17 ldw r3,-20(fp) 363: e12ffb17 ldw r3,-20(fp) 364: e0bffe15 stw r2,-8(fp) 374: e0fff17 ldw r3,-20(fp) 364: e0fff17 ldw r3,-20(fp) 365: e0ff617 ldw r3,-20(fp) 365: e0ff617 ldw r3,-20(fp) 366: e0ff617 ldw r3,-20(fp) 375: e17fb17 ldw r3,-20(fp) 376: e17fb17 ldw r3,-20(fp) 376: e17fb17 ldw r3,-20(fp) 3		2d8:	e0tttd17	ldw	r3,-12((tp)
2e0: $10C7t972$ custom 229,r3,r2,r3 2e4: $0005f932$ custom 228,r2,zero,zero 2e8: $e0bffe15$ stw r2,-8(fp) 2ec: $e0fff15$ stw r3,-4(fp) R = A - B; 2f0: $e13ffa17$ ldw r4,-24(fp) 2f4: $e17ffb17$ ldw r5,-20(fp) 2f8: $e0bffc17$ ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: $10C7f9b2$ custom 230,r3,r2,r3 308: $0005f932$ custom 230,r3,r2,r3 308: $0005f932$ custom 228,r2,zero,zero 30c: $e0bffe15$ stw r3,-4(fp) R = A * B; 314: $e13ffa17$ ldw r4,-24(fp) 310: $e0fff17$ ldw r5,-20(fp) 320: $e0fff17$ ldw r5,-20(fp) 310: $e0fff17$ ldw r5,-20(fp) 310: $e0fff17$ ldw r3,-12(fp) 320: $e0fff15$ stw r2,-8(fp) 331: $e0fff15$ stw r2,-8(fp) 332: $e0fff15$ stw r2,-8(fp) 334: $e0fff15$ stw r2,-8(fp) 335: $e13ffa17$ ldw r4,-24(fp) 336: $e0bffe15$ stw r2,-8(fp) 341: $e0fff15$ stw r3,-4(fp) R = A / B; 338: $e13ffa17$ ldw r4,-24(fp) 330: $e0bffe15$ stw r2,-8(fp) 341: $e0fff17$ ldw r3,-12(fp) 342: $2141f832$ custom 228,r2,zero,zero 330: $e0bffe15$ stw r2,-8(fp) 341: $e0fff15$ stw r3,-4(fp) R = A / B; 338: $e13ffa17$ ldw r4,-24(fp) 340: $e0bffe17$ ldw r3,-12(fp) 341: $e0fff17$ ldw r3,-12(fp) 342: $2141f832$ custom 228,r2,zero,zero 350: $0005f932$ custom 228,r2,zero,zero 351: $e0bffe17$ ldw r3,-12(fp) 342: $2141f832$ custom 224,zero,r4,r5 342: $10C7fa32$ custom 224,zero,r4,r5 342: $10C7fa32$ custom 232,r3,r2,r3 350: $0005f932$ custom 232,r3,r2,r4(fp) 358: $e0fff15$ stw r3,-4(fp)		2dc:	2141†832	custor	n	224,zero,r4,r5
2e4: 0005f932 custom 228,r2,zero,zero 2e8: e0bffe15 stw r2,-8(fp) 2ec: e0ffff15 stw r3,-4(fp) R = A - B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f8: e0bffc17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e0bffe15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 310: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-20(fp) 312: e0bffc17 ldw r3,-20(fp) 320: e0fff17 ldw r3,-21(fp) 320: e0fff17 ldw r3,-21(fp) 321: e0fff17 ldw r3,-12(fp) 322: 0005f932 custom 231,r3,r2,r3 328: 10c7f9f2 custom 231,r3,r2,r3 320: e0fff15 stw r2,-8(fp) 321: e0fff15 stw r2,-8(fp) 322: 0005f932 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 344: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 340: e0bffe17 ldw r3,-12(fp) 341: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 340: e0bffe17 ldw r3,-12(fp) 341: e0fff17 ldw r3,-20(fp) 342: 2141f832 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 341: e0fff17 ldw r3,-20(fp) 342: e0fff17 ldw r3,-20(fp) 343: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r3,-20(fp) 340: e0bffe17 ldw r3,-20(fp) 340: e0bffe15 stw r2,-8(fp) 341: e0fff17 ldw r3,-20(fp) 342: e0fff17 ldw r3,-20(fp) 343: e0fff17 ldw r3,-20(fp) 344: e0fff17 ldw r3,-20(fp) 345: e0fff17 ldw r3,-20(fp) 346: 2141f832 custom 232,r3,r2,r3 350: 0005f932 custom 232,r3,r2,r3 3		2e0:	10c7f972	custor	n	229,r3,r2,r3
2e8:e0bffe15stwr2,-8(fp)2ec:e0fff15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} - \mathbf{B};$ 2f0:e13ffa17ldwr4,-24(fp)2f4:e17ffb17ldwr5,-20(fp)2f8:e0bffc17ldwr3,-12(fp)300:2141f832custom224,zero,r4,r5304:1007f9b2custom230,r3,r2,r3308:0005f932custom228,r2,zero,zero30c:e0bffe15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} * \mathbf{B};$ 314:e13ffa17ldw310:e0fff15stwr3,-2(fp)312:e0bffc17ldwr4,-24(fp)318:e17ffb17ldwr5,-20(fp)310:e0fff17ldwr3,-12(fp)324:2141f832custom224,zero,r4,r5328:1007f9f2custom231,r3,r2,r3300:e0bffe15stwr2,-8(fp)324:2141f832custom224,zero,zero330:e0bffe15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} / \mathbf{B};$ 338:e13ffa17330:e0bffe15stwr3,-2(fp)342:e0fff17ldwr4,-24(fp)343:e0fff17ldwr4,-24(fp)344:e0fff17ldwr4,-24(fp)350:0005f932custom224,zero,r4,r5361:e0fff17ldwr3,-12(fp)362:e0fff17ldwr4,-24(fp)363:e0fff17ldwr4,-24(fp) <td< td=""><td></td><td>2e4:</td><td>0005†932</td><td>custor</td><td>n</td><td>228,r2,zero,zero</td></td<>		2e4:	0005†932	custor	n	228,r2,zero,zero
2ec: e0fff15 stw r3,-4(fp) R=A-B; 2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f6: e0bffc17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10C7f9b2 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R=A*B; 314: e13ffa17 ldw r4,-24(fp) 312: e0bffc17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 322: 2141f832 custom 234,zero,r4,r5 328: 10C7f9f2 custom 231,r3,r2,r3 320: e0bffe15 stw r2,-8(fp) 321: e0bffe15 stw r2,-8(fp) 322: e0fff17 ldw r3,-12(fp) 324: 2141f832 custom 231,r3,r2,r3 32c: 0005f932 custom 231,r3,r2,r3 32c: 0005f932 custom 232,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R=A/B; 338: e13ffa17 ldw r4,-24(fp) 330: e0bffe15 stw r2,-8(fp) 340: e0bffe17 ldw r3,-12(fp) 341: e0fff17 ldw r3,-12(fp) 342: 2141f832 custom 232,r3,r2,r3 350: 0005f932 custom 232,r3,r2,r3 350: 00		2e8:	e0bffe15	stw	r2,-8(t	p)
R=A-B; 2f0: $e13ffa17$ ldw r4,-24(fp) 2f4: $e17ffb17$ ldw r5,-20(fp) 2f8: $e0bffc17$ ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 300: 2141f832 custom 230,r3,r2,r3 300: 200: 2141f832 custom 230,r3,r2,r3 300: 0005f932 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 301: $e0fff15$ stw r2,-8(fp) 310: 310: $e0fff15$ stw r3,-4(fp) R=A*B; 314: $e13ffa17$ ldw r4,-24(fp) 318: $e17ffb17$ ldw r4,-24(fp) 318: $e17ffb17$ ldw r4,-24(fp) 312: $e00ffc17$ ldw r4,-24(fp) 322: $e00ffc17$ ldw r4,-24(fp) 324: $2141f832$ custom 224,zero,r4,r5 328: $10c7f92$ custom 231,r3,r2,r3 320: $e00ffc15$ stw r2,-8(fp) 334: $e01ff17$		2ec:	e0###15	stw	r3,-4(t	p)
2f0: e13ffa17 ldw r4,-24(fp) 2f4: e17ffb17 ldw r5,-20(fp) 2f6: e0fffd17 ldw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e0bffc17 ldw r3,-12(fp) 320: e0bffe15 stw r2,-16(fp) 320: e0fff15 stw r3,-4(fp) 321: e0bffc17 ldw r3,-12(fp) 322: 0005f932 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 320: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 332: 0005f932 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 320: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 336: e17ffb17 ldw r5,-20(fp) 340: e0bffc17 ldw r5,-20(fp) 340: e0bffc17 ldw r3,-12(fp) 341: e0fffd17 ldw r3,-12(fp) 342: 2141f832 custom 232,r3,r2,r3 350: 0005f932 custom 232,r3,r2,r4(fp)		R = A - B	;			
2f4: e17fb17 ldw r5,-20(fp) 2f8: e0bffc17 ldw r2,-16(fp) 300: 2141f832 custom 230,r3,r2,r3 304: 10c7f9b2 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 ldw r4,-24(fp) 318: e17ffb17 ldw r5,-20(fp) 310: e0fff17 ldw r5,-20(fp) 310: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 320: e0fff17 ldw r3,-12(fp) 321: e0bffe15 stw r2,-8(fp) 322: 0005f932 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 32c: 0005f932 custom 228,r2,zero,zero 330: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 ldw r4,-24(fp) 330: e0bffe17 ldw r3,-12(fp) 341: e0fff17 ldw r3,-12(fp) 342: 2141f832 custom 224,zero,r4,r5 351: e13ffa17 ldw r4,-24(fp) 362: e17ffb17 ldw r4,-24(fp) 363: e13ffa17 ldw r4,-24(fp) 361: e0bffc17 ldw r3,-12(fp) 361: e0ffc17 ldw r3,-12(fp) 361: e0fffc17 ldw r3,-12(fp) 361: e0ffc17 ldw r3,-12(fp) 362: e17fb17 ldw r3,-12(fp) 363: e17fb17 ldw r3,-12(fp) 364: e0fff15 stw r3,-4(fp)		2f0:	e13ffa17	ldw	r4,-24((fp)
2f8: e0bffc17 dw r2,-16(fp) 2fc: e0fffd17 dw r3,-12(fp) 300: 2141f832 custom 224,zero,r4,r5 304: 10c7f9b2 custom 230,r3,r2,r3 308: 0005f932 custom 228,r2,zero,zero 30c: e0bffe15 stw r2,-8(fp) 310: e0fff15 stw r3,-4(fp) R = A * B; 314: e13ffa17 dw r4,-24(fp) 316: e0bffc17 dw r5,-20(fp) 310: e0fff17 dw r5,-20(fp) 320: e0fffd17 dw r3,-12(fp) 320: e0fffd17 dw r3,-12(fp) 324: 2141f832 custom 224,zero,r4,r5 328: 10c7f9f2 custom 231,r3,r2,r3 320: e0bffe15 stw r2,-8(fp) 334: e0fff15 stw r3,-4(fp) R = A / B; 338: e13ffa17 dw r4,-24(fp) 332: e17ffb17 dw r4,-24(fp) 334: e0fff15 stw r2,-8(fp) 335: e17ffb17 dw r4,-24(fp) 346: e0fff17 dw r4,-24(fp) 347: e0fff17 dw r4,-24(fp) 348: e13ffa17 dw r4,-24(fp) 340: e0bffc17 dw r3,-12(fp) 341: e0fff17 dw r3,-12(fp) 342: e0fff17 dw r4,-24(fp) 343: e13ffa17 dw r4,-24(fp) 344: e0fff17 dw r3,-12(fp) 345: e17ffb17 dw r3,-12(fp) 346: e0bffc17 dw r3,-12(fp) 347: e0fff17 dw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 340: 0005f932 custom 232,r3,r2,r3 350: 0005f932 custom 232,r3,r3,r2,r3 350: 0005f932 custom 232,r3,r3,r2,r3 350: 0005f932 custom 232,r3,r3,r3,r3,r3,r3,r3,r3,r		2f4:	e17ffb17	ldw	r5,-20((fp)
$\begin{array}{ccccc} 2fc: & eOfffd17 & ldw r3,-12(fp) \\ 300: & 2141f832 & custom & 224,zero,r4,r5 \\ 304: & 10c7f9b2 & custom & 228,r2,zero,zero \\ 30c: & eObffe15 & stw r2,-8(fp) \\ 310: & eOfff15 & stw r3,-4(fp) \\ \hline \textbf{R}=\textbf{A}*\textbf{B}; \\ 314: & e13ffa17 & ldw r4,-24(fp) \\ 318: & e17ffb17 & ldw r5,-20(fp) \\ 310: & eOfff17 & ldw r3,-12(fp) \\ 320: & eOfff15 & stw r2,-8(fp) \\ 330: & eObffe15 & stw r2,-8(fp) \\ 334: & eOfff15 & stw r3,-4(fp) \\ \hline \textbf{R}=\textbf{A}/\textbf{B}; \\ 338: & e13ffa17 & ldw r4,-24(fp) \\ 330: & eObffc17 & ldw r3,-12(fp) \\ 340: & eObffc17 & ldw r3,-12(fp) \\ 344: & eOfffd17 & ldw r3,-12(fp) \\ 344: & eOfffd17 & ldw r3,-12(fp) \\ 344: & eOfffd17 & ldw r3,-12(fp) \\ 346: & 2141f832 & custom & 224,zero,r4,r5 \\ 34c: & 10c7fa32 & custom & 224,zero,r4,r5 \\ 34c: & 10c7fa32 & custom & 224,zero,r4,r5 \\ 350: & 0005f932 & custom & 228,r2,zero,zero \\ 354: & eObffe15 & stw r2,-8(fp) \\ 358: & eOfff15 & stw r3,-4(fp) \\ \end{array}$		2f8:	e0bffc17	ldw	r2,-16((fp)
300: $2141f832$ custom 224 ,zero,r4,r5 $304:$ $10C7f9b2$ custom 230 ,r3,r2,r3 $308:$ $0005f932$ custom 228 ,r2,zero,zero $30c:$ $e0bffe15$ stw $r3,-4(fp)$ $310:$ $e0fff15$ stw $r3,-4(fp)$ $R = A * B;$ 8 $314:$ $e13ffa17$ ldw $r4,-24(fp)$ $318:$ $e17ffb17$ ldw $r5,-20(fp)$ $31c:$ $e0bffc17$ ldw $r3,-12(fp)$ $320:$ $e0fff17$ ldw $r3,-12(fp)$ $324:$ $2141f832$ custom 224 ,zero,r4,r5 $328:$ $10C7f9f2$ custom $231,r3,r2,r3$ $32c:$ $0005f932$ custom $223,r2,r2,r2,r2,r2,r2,r2,r2,r3$ $330:$ $e0bffe15$ stw $r2,-8(fp)$ $334:$ $e0fff17$ ldw $r4,-24(fp)$ $336:$ $e13ffa17$ ldw $r4,-24(fp)$ $336:$ $e0fff15$ stw $r3,-4(fp)$ $R = A / B;$ $336:$ $e17ffb17$ ldw $340:$ $e0fff17$ ldw $r4,-24(fp)$ $344:$ $e0fff17$ ldw $r3,-12(fp)$ $344:$ $e0fff17$ ldw $r3,-12(fp)$ $344:$ $e0fff17$ ldw $r3,-12(fp)$ $346:$ $2141f832$ custom 224 ,zero,r4,r5 $34c:$ $10C7fa32$ custom 224 ,zero,r4,r5 $350:$ $0005f932$ custom 224 ,zero,r4,r5 $351:$ $e0fff15$ stw $r2,-8(fp)$ $358:$ $e0$		2fc:	e0fffd17	ldw	r3,-12((fp)
$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$		300:	2141f832	custor	n	224,zero,r4,r5
308:0005f932custom228,r2,zero,zero30c:e0bffe15stwr2,-8(fp)310:e0fff15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} * \mathbf{B}$;314:e13ffa17ldw318:e17ffb17ldwr4,-24(fp)318:e17ffb17ldwr5,-20(fp)310:e0bffc17ldwr3,-12(fp)320:e0fffc17ldwr3,-12(fp)321:2141f832custom224,zero,r4,r5328:10c7f9f2custom231,r3,r2,r3320:e0bffe15stwr2,-8(fp)334:e0fff15stwr2,-8(fp)334:e0fff15stwr3,-4(fp)R = A / B;388:e13ffa17ldw336:e13ffa17ldwr4,-24(fp)341:e0fff17ldwr5,-20(fp)342:2141f832custom224,zero,r4,r5343:e0fff17ldwr3,-12(fp)344:e0fff17ldwr3,-12(fp)345:2141f832custom224,zero,r4,r5346:10c7fa32custom224,zero,zero350:0005f932custom228,r2,zero,zero351:e0bffe15stwr2,-8(fp)358:e0fff15stwr3,-4(fp)		304:	10c7f9b2	custor	n	230,r3,r2,r3
$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$		308:	0005f932	custor	n	228,r2,zero,zero
310:eOffff15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} * \mathbf{B};$ 314:e13ffa17ldwr4,-24(fp)318:e17ffb17ldwr5,-20(fp)31c:e0bffc17ldwr2,-16(fp)320:e0fffd17ldwr3,-12(fp)324:2141f832custom224,zero,r4,r5328:10c7f9f2custom231,r3,r2,r332c:0005f932custom228,r2,zero,zero330:e0bffe15stwr2,-8(fp)334:e0fff15stwr3,-4(fp) $\mathbf{R} = \mathbf{A} / \mathbf{B};$ 338:e13ffa17340:e0bffc17ldwr4,-24(fp)342:2141f832custom224,zero,r4,r5343:e0fffd17ldwr3,-12(fp)344:e0fffd17ldwr3,-12(fp)348:2141f832custom224,zero,r4,r534c:10c7fa32custom224,zero,zero350:0005f932custom228,r2,zero,zero354:e0bffe15stwr2,-8(fp)358:e0fff15stwr3,-4(fp)		30c:	e0bffe15	stw	r2,-8(t	p)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$		310:	e0###15	stw	r3,-4(t	p)
314:e13fta17ldwr4,-24(tp)318:e17ffb17ldwr5,-20(tp)31c:e0bffc17ldwr5,-20(tp)320:e0fffd17ldwr3,-12(tp)324:2141f832custom224,zero,r4,r5328:10c7f9f2custom231,r3,r2,r332c:0005f932custom228,r2,zero,zero330:e0bffe15stwr2,-8(tp)334:e0fff15stwr3,-4(tp)R=A/B;338:e13ffa17ldw330:e0bffc17ldwr4,-24(tp)332:e17ffb17ldwr5,-20(tp)340:e0bffc17ldwr5,-20(tp)342:2141f832custom224,zero,r4,r5342:10c7fa32custom224,zero,r4,r5342:10c7fa32custom232,r3,r2,r3350:0005f932custom228,r2,zero,zero354:e0bffe15stwr2,-8(tp)358:e0fff15stwr3,-4(tp)		R = A * E	3;			
318: $e1/thb17$ Idw $r5,-20(tp)$ 31c: $e0bffc17$ Idw $r2,-16(fp)$ 320: $e0fffd17$ Idw $r3,-12(fp)$ 324:2141f832 $custom$ 224,zero,r4,r5 328: $10c7f9f2$ $custom$ 231,r3,r2,r3 32c: $0005f932$ $custom$ 228,r2,zero,zero 330: $e0bffe15$ stw $r2,-8(fp)$ 334: $e0fff15$ stw $r3,-4(fp)$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $338:$ $e13ffa17$ Idw338: $e13ffa17$ Idw $r4,-24(fp)$ 336: $e17ffb17$ Idw $r5,-20(fp)$ 340: $e0bffc17$ Idw $r3,-12(fp)$ 344: $e0fffd17$ Idw $r3,-12(fp)$ 348:2141f832 $custom$ 224,zero,r4,r5 34c: $10c7fa32$ $custom$ 224,zero,zero 350: $0005f932$ $custom$ 228,r2,zero,zero 354: $e0bffe15$ stw $r2,-8(fp)$ 358: $e0fff15$ stw $r3,-4(fp)$		314:	e13tta17	ldw	r4,-24((tp)
31C: $e0bftc17$ Idw $r2,-16(tp)$ 320: $e0ffd17$ Idw $r3,-12(tp)$ 324: $2141f832$ $custom$ $224,zero,r4,r5$ 328: $10c7f9f2$ $custom$ $231,r3,r2,r3$ 32c: $0005f932$ $custom$ $228,r2,zero,zero$ 330: $e0bffe15$ stw $r2,-8(tp)$ 334: $e0ffff15$ stw $r3,-4(tp)$ $R = A/B;$ $338:$ $e13ffa17$ Idw 330: $e0bffc17$ Idw $r4,-24(tp)$ 331: $e17ffb17$ Idw $r5,-20(tp)$ 340: $e0bffc17$ Idw $r5,-20(tp)$ 344: $e0fffd17$ Idw $r3,-12(tp)$ 348: $2141f832$ $custom$ $224,zero,r4,r5$ 34c: $10c7fa32$ $custom$ $224,zero,r4,r5$ 350: $0005f932$ $custom$ $228,r2,zero,zero$ 354: $e0bffe15$ stw $r2,-8(tp)$ 358: $e0fff15$ stw $r3,-4(tp)$		318:	e17ttb17	ldw	r5,-20((fp)
$320:$ e0fftd17ldwr3,-12(tp) $324:$ 2141832 custom $224,zero,r4,r5$ $328:$ $10c7f9f2$ custom $231,r3,r2,r3$ $32c:$ $0005f932$ custom $228,r2,zero,zero$ $330:$ $e0bffe15$ stw $r2,-8(fp)$ $334:$ $e0fff15$ stw $r3,-4(fp)$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $\mathbf{R} = 13ffa17$ $330:$ $e13ffa17$ ldw $r4,-24(fp)$ $33c:$ $e13ffa17$ ldw $r5,-20(fp)$ $340:$ $e0bffc17$ ldw $r5,-20(fp)$ $340:$ $e0bffc17$ ldw $r3,-12(fp)$ $344:$ $e0fffd17$ ldw $r3,-12(fp)$ $348:$ $2141f832$ custom $224,zero,r4,r5$ $34c:$ $10c7fa32$ custom $232,r3,r2,r3$ $350:$ $0005f932$ custom $228,r2,zero,zero$ $354:$ $e0bffe15$ stw $r2,-8(fp)$ $358:$ $e0fff15$ stw $r3,-4(fp)$		310:	eUbttc17	Idw	r2,-16	(tp)
3.24 : 21411832 custom 224 ,zero,r4,r5 328 : $10c79f2$ custom 231 ,r3,r2,r3 $32c$: $0005f932$ custom 228 ,r2,zero,zero 330 : $e0bffe15$ stw $r2,-8(fp)$ 334 : $e0fff15$ stw $r3,-4(fp)$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ 338 : $e13ffa17$ ldw $r4,-24(fp)$ $33c$: $e17ffb17$ ldw $r5,-20(fp)$ 340 : $e0bffc17$ ldw $r3,-12(fp)$ 344 : $e0fffd17$ ldw $r3,-12(fp)$ 348 : $2141f832$ custom 224 ,zero,r4,r5 $3c$: $10c7fa32$ custom 222 ,r3,r2,r3 350 : $0005f932$ custom 228 ,r2,zero,zero 354 : $e0bffe15$ stw $r2,-8(fp)$ 358 : $e0fff15$ stw $r3,-4(fp)$		320:	e0md17	Idw	r3,-12((tp)
3.28 : $10c/rH2$ custom $231,r3,r2,r3$ $32c$: $0005f932$ $custom$ $228,r2,zero,zero$ 330 : $e0bffe15$ stw $r2,-8(fp)$ 334 : $e0fff15$ stw $r3,-4(fp)$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ $\mathbf{R} = \mathbf{A} / \mathbf{B};$ 336 : $e13ffa17$ ldw $r4,-24(fp)$ $33c$: $e17ffb17$ ldw $r5,-20(fp)$ 340 : $e0bffc17$ ldw $r2,-16(fp)$ 344 : $e0fffd17$ ldw $r3,-12(fp)$ 348 : $2141f832$ $custom$ $224,zero,r4,r5$ $34c$: $10c7fa32$ $custom$ $222,r3,r2,r3$ 350 : $0005f932$ $custom$ $228,r2,zero,zero$ 354 : $e0bffe15$ stw $r2,-8(fp)$ 358 : $e0fff15$ stw $r3,-4(fp)$		324:	21411832	custor	n	224,zero,r4,r5
32C: 0005r932 Custom 228_r2_z reo,zero 330: e0bffe15 stw $r2_z$ -8(fp) 334: e0fff15 stw $r3_z$ -4(fp) R = A / B; 338: e13ffa17 ldw $r4_z$ -24(fp) 332: e17ffb17 ldw $r4_z$ -24(fp) 340: e0bffc17 ldw $r5_z$ -20(fp) 340: e0bffc17 ldw $r3_z$ -12(fp) 344: e0fffd17 ldw $r3_z$ -12(fp) 348: 2141f832 Custom 224_zero,r4,r5 34c: 10c7fa32 custom 224_zero,zero,zero 350: 0005f932 custom 228_r2_zero,zero 358: e0fff15 stw $r3_z$ -4(fp)		328:	100/1912	custor	n	231,r3,r2,r3
3-30: e00ffe15 stw r2,-8(tp) 334: e0fff15 stw r3,-4(tp) R=A/B; 338: e13ffa17 ldw r4,-24(fp) 33c: e13ffa17 ldw r4,-24(fp) 33c: e13ffa17 ldw r5,-20(fp) 340: e0bffc17 ldw r5,-20(fp) 344: e0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		32C:	00051932	custor	n	228,r2,zero,zero
334: e0fff15 stw r3,-4(tp) R=A/B; 338: e13ffa17 ldw r4,-24(fp) 33c: e17ffb17 ldw r5,-20(fp) 340: e0bffc17 ldw r2,-16(fp) 344: e0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		330:	eUbffe15	stw	r2,-8(f	p)
R = A / B; 338: e13ffa17 ldw r4,-24(fp) 33c: e17ffb17 ldw r5,-20(fp) 340: e0bffc17 ldw r2,-16(fp) 344: e0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15		334:	eomiiis	Stw	r3,-4(†	p)
33c: e17ffb17 ldw r5,-20(fp) 340: e0bffc17 ldw r2,-16(fp) 344: e0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		κ=A/E 338:	e13ffa17	ldw	r424	(fp)
340: c0bffc17 ldw r2,-16(fp) 344: c0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		330	e17ffb17	ldw	r520	(fp)
344: e0fffd17 ldw r3,-12(fp) 348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		340.	e0bffc17	ldw	r2-16	(fp)
348: 2141f832 custom 224,zero,r4,r5 34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		344	e0fffd17	ldw	r3-12	(fp)
34c: 10c7fa32 custom 232,r3,r2,r3 350: 0005f932 custom 228,r2,zero,zero 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		348.	2141f832	custor	n, 12(224.zero r4 r5
350: 0005f932 custom 228,r3,rs,r3 354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		340	10c7fa32	custor	n	232.r3.r2 r3
354: e0bffe15 stw r2,-8(fp) 358: e0fff15 stw r3,-4(fp)		350	0005f932	custor	n	228.r2.zero zero
358: e0ffff15 stw r3,-4(fp)		354	e0bffe15	stw	r2 -8(f	n)
		250.	e0ffff15	stw	r34(f	n)
		220.	CONTRACT	3644		N/



- 3-7. 整数(32bit)から倍精度浮動小数への型変換のシーケンス
 - 1. 入力の 32bit 整数データをカスタム・インストラクション・モジュールに入力し、演算完了を待ち、演算結果 の上位 32bit を出力します。



2. カスタム・インストラクション・モジュールから演算結果の下位 32bit を出力して演算結果 64bit を合成しま す。



【C 言語記述とアセンブラ・コードの例】

{ int A = 123456; volatile double R; R = (double)A;	{ int A = 123456; 29c: 008000b4 movhi r2,2 2a0: 10b89004 addi r2,r2,-7616 2a4: e0bffd15 stw r2,-12(fp) volatile double R;
	R = (double)A; 2a8: e0bffd17 ldw r2,-12(fp) 2ac: 1007fa72 custom 233,r3,r2,zero 2b0: 0005f932 custom 228,r2,zero,zero 2b4: e0bffe15 stw r2,-8(fp) 2b8: e0fff15 stw r3,-4(fp)



3-8. 倍精度浮動小数⇒整数(32bit)への型変換のシーケンス

1. 入力の 64bit データを、2 つのレジスタを使ってカスタム・インストラクション・モジュールに入力し、演算完 了を待ち、演算結果の下位 32bit を出力します。



【C 言語記述とアセンブラ・コードの例】





3-9. 比較演算のシーケンス

1. 入力 A の 64bit データを、2 つのレジスタを使ってカスタム・インストラクション・モジュールに入力し、モジ ュール内部のレジスタに 64bit データを保持します。



2. 入力 B の 64bit データを、2 つのレジスタを使ってカスタム・インストラクション・モジュールに入力し、演算 完了を待ち、条件が成立していれば "0" を、不成立なら "1" を最下位 bit に設定して出力します。

入力 B:	上位 32bit(r4)	下位 32bit(r5)
アセンブラ・	⊐ード: custom	X, <u>r3,r4,r5</u> X, <u>r3,r4,r5</u> 人 カスタム・インストラクション・モジュール
演算結果:	(r3)	●──── 最下位 bit(r3) : 0=条件成立, 1=条件不成立

{

【C 言語記述とアセンブラ・コードの例】

{

double A = 1.23456789; double B = 1.23456789; volatile int C = 0;

if(A == B) C ++; if(A < B) C ++; if(A <= B) C ++;

	A = 1.23456	789;	
29c:	0090a134	movhi	r2,17028
2a0:	10b786c4	addi	r2,r2,-8677
2a4:	e0bffb15	stw	r2,-20(fp)
2a8:	008ffd34	movhi	r2,16372
2ac:	10b03284	addi	r2,r2,-16182
2b0:	e0bffc15	stw	r2,-16(fp)
double	B = 1.23456	789;	
2b4:	0090a134	movhi	r2,17028
2b8:	10b786c4	addi	r2,r2,-8677
2bc:	e0bffd15	stw	r2,-12(fp)
2c0:	008ffd34	movhi	r2,16372
2c4:	10b03284	addi	r2,r2,-16182
2c8:	e0bffe15	stw	r2,-8(fp)
volatile	int C=0;		
2cc:	e03fff15	stw	zero,-4(fp)
f(A == P	s) C ++:		
2d0:	e13ffb17	ldw	r420(fp)
2d4:	e17ffc17	ldw	r516(fp)
2d8.	e0bffd17	ldw	r2 -12(fn)
2dc	e0fffe17	ldw	r3 -8(fn)
2e0.	2141f832	custor	n 774 zero r4 r5
2e0:	10c5fc32	custor	n 240.r2.r2.r3
2e8:	1000031e	bne	r2 zero 2f8 <main+0x68></main+0x68>
2ec:	e0bfff17	ldw	r24(fn)
2f0.	10800044	addi	r2 r2 1
2f4.	e0bfff15	stw	r2 -4(fn)
f(A < B)	C++:		
2f8:	e13ffb17	ldw	r420(fp)
2fc:	e17ffc17	ldw	r516(fp)
300:	e0bffd17	ldw	r212(fp)
304:	e0fffe17	ldw	r38(fp)
308:	2141f832	custor	n 224.zero.r4.r5
30c:	10c5faf2	custor	n 235,r2.r2.r3
310:	10000326	bea	r2.zero.320 <main+0x90< td=""></main+0x90<>
314:	e0bfff17	ldw	r2,-4(fp)
318:	10800044	addi	r2.r2.1
31c:	e0bfff15	stw	r2,-4(fp)
f(A <= B	B) C ++:		
320:	e13ffb17	ldw	r4,-20(fp)
324:	e17ffc17	ldw	r5,-16(fp)
328:	e0bffd17	ldw	r2,-12(fp)
	e0fffe17	ldw	r3,-8(fp)
32c:		au at au	n 224,zero,r4.r5
32c: 330:	2141f832	custor	, , ,
32c: 330: 334:	2141f832 10c5fb32	custor	n 236,r2,r2,r3
32c: 330: 334: 338:	2141f832 10c5fb32 10000326	custor custor bea	n 236,r2,r2,r3 r2,zero,348 <main+0xb8></main+0xb8>
32c: 330: 334: 338: 33c:	2141f832 10c5fb32 10000326 e0bfff17	custor custor beq ldw	n 236,r2,r2,r3 r2,zero,348 <main+0xb8 r2,-4(fp)</main+0xb8
32c: 330: 334: 338: 33c: 340:	2141f832 10c5fb32 10000326 e0bfff17 10800044	custor beq ldw addi	n 236,r2,r2,r3 r2,zero,348 <main+0xb8 r2,-4(fp) r2,r2,1</main+0xb8



4. <u>実装</u>

4-1. ベース・プロジェクトの準備

Nios II が動作する Quartus Prime プロジェクトをあらかじめ用意してください。なお、本資料では Verilog HDL で実装していますが VHDL での実装も可能です。

4-2. IP Catalog を使用した倍精度浮動小数演算 HDL の生成

4-2-1. 倍精度浮動小数加減算コアの生成

 Quartus Prime の IP Catalog ウィンドウの Installed IP ⇒ Library ⇒ Basic Function ⇒ Arithmetic ⇒ ALTERA_FP_FUNCTION を選択します。ダブルクリックか[Add]をクリックし、 IP variation file name に生成 するファイルパスおよびファイル名、 IP variation file type に HDL タイプを選択し[OK]をクリックします。 本実装では、生成するファイル名を "double_add.v"、Verilog HDL で生成します。





2. ALTERA_FP_FUNCTION の IP コア生成ダイアログ・ボックスが表示されます。 Function タブを下記の設 定にします。

ALTERA FP FUNCTIONS - double add	
ALTERA_FP_FUNCTIONS	Function
MogeCore altera_fp_functions	Family : <u>Arithmetic</u>
Block Diagram Functionality Performance	Name : <u>Add/Sub</u>
Function	Use Select Signal : <u>ON</u>
double_add Arithmetic Name: Add/Sub	Floating Point Data
areset	Format : <u>Double</u>
en conduit Floating Point Data	Rounding
aconduit	Relax rounding : OFF
conduit Rounding	
opsei conduit Mode: nearest with tie breaking to even	Ports
altera_fp_functions	Generate an enable port : <u>ON</u>
Generate an enable port	
Info: double_add: C:/intelipga/17.1/quartus//dspba/backend/windows64/cmdPolyEval-target Cyclo Info: double_add: C:/intelipga/17.1/quartus//dspba/backend/windows64/cmdPolyEval-target Cyclo	nev -frequency 200 -name r neV -frequency 200 -name r
6	•
	Cancel Finish

3. Performance タブに切替え、Target に要求される動作周波数やレイテンシを設定すると大凡のリソース 使用量やレイテンシが Report に表示されます。本実装では Goal を Combined とし、ターゲット周波 数を 100MHz、レイテンシを 9 cycles に設定します。

Block Diagram Show signals	Functionality	Performance	,	Target : <u>100</u> MHz
double_add clk clock conduit areset en conduit a conduit b conduit opSel conduit altera_fp_functions	Goal: Target: Target: * Report Minimum ache Resource Esti Multiplies: LUTs: Memory Bits: Memory Blocks	Combined 100 9 eivable latency at given mates: 0 1897 0 5: 0	MHz cycles	es, register padding will o
Info: double_add: C:/intelfpga/17.1 Info: double_add: C:/intelfpga/17.1	1/quartus//dspba/ba 1/quartus//dspba/ba	ckend/windows64/cm ckend/windows64/cm	dPolyEval -target C dPolyEval -target C	ycloneV -frequency 100 -name r ycloneV -frequency 100 -name r Cancel Finish



4. [Finish] をクリックし、Generation ダイアログ・ボックスに Generation Successful と表示されたら [Exit] をクリックして閉じます。



5. 生成された IP コアを Quartus Prime プロジェクトに組み込むかどうかのダイアログが表示されますの で、[Yes] をクリックして組み込みます。





4-2-2. 倍精度浮動小数乗算コアの生成

- 1. 前項(4-2-1)の 1. と同様の操作で、 IP Catalog から ALTERA_FP_FUNCTION を開き、生成するファイル名 を "double_mul.v"、Verilog HDL で生成します。
- 2. ALTERA_FP_FUNCTION の IP コア生成ダイアログ・ボックスが表示されます。 Function タブを下記の設 定にします。

ALTERA_FP_FUNCTIONS = double_mul	Function
MegaCore' altera_fp_functions	Family : <u>Arithmetic</u>
Block Diagram Show signals	Name : <u>Multiply</u>
Function	Floating Point Data
double_mul Family: Arithmetic V Name: Multiply V	Format : <u>Double</u>
Clk clock conduit	Rounding
en conduit Format Double	Relax rounding : <u>OFF</u>
a conduit TRounding	Ports
b conduit attera_fp_functions Mode: nearest with tie breaking to even Relax rounding to round up or down to reduce resource of	Generate an enable port : <u>ON</u>
▼ Ports	
Generate an enable port	
Info: double_mul: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target Cyclone Info: double_mul: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target Cyclone Info: double_mul: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target Cyclone	eV -frequency 100 -name none -noCha eV -frequency 100 -name none -noCha
	Cancel Finish

3. Performance タブに切替え、Target に要求される動作周波数やレイテンシを設定すると大凡のリソース 使用量やレイテンシが Report に表示されます。本実装では Goal を Combined とし、ターゲット周波 数を 100MHz、レイテンシを 5 cycles に設定します。

how signals	Functionality F	Performance		1	Farget : <u>100</u> MHz Farget : <u>5</u> cycles
clock conduit reset conduit conduit conduit altera_fp_functions	Target: Target: Report Minimum acheiv Resource Estim Multiplies: LUTs: Memory Bits:	able latency at given at es:	MHz cycles	cycles, register pade	ding will occur to meet late
o: double_mul: C:/intelfpga/17.1/q	Memory Blocks:	0 cend/windows64/c	mdPolyEval -targe	t CycloneV -frequen	cy 100 -name none -noCha



- 4. [Finish] をクリックし、Generation ダイアログ・ボックスに Generation Successful と表示されたら [Exit] をクリックして閉じます。
- 5. 生成された IP コアを Quartus Prime プロジェクトに組み込むかどうかのダイアログが表示されますので、 [Yes] をクリックして組み込みます。

4-2-3. 倍精度浮動小数除算コアの生成

- 1. 前項の 1. と同様の操作で、 IP Catalog から ALTERA_FP_FUNCTION を開き、生成するファイル名 を "double_div.v"、Verilog HDL で生成します。
- 2. ALTERA_FP_FUNCTION の IP コア生成ダイアログ・ボックスが表示されます。 Function タブを下記の設 定にします。

Heyeron altera_fp_functions Block Diagram Show signals double_div clik clock conduit areset reset en conduit b conduit attera_fp_functions	Functionality Performance Function Function Family: Arithmetic Name: Divide Ploating Point Data Image: Comparison of the provided of the pro	Name : <u>Divide</u> Floating Point Data Format : <u>Double</u> Rounding Relax rounding : <u>OFF</u> Algorithm Method : <u>Polynomial</u> Ports Generate an enable port : <u>ON</u>
 Info: double_div: C:/intel/pga/17.1/ Info: double_div: C:/intel/pga/17.1/ Info: double_div: C:/intel/pga/17.1/ 	Generate an enable port	eV -frequency 100 -name none -noChanVal eV -frequency 100 -name none -noChanVal eV Cancel Finish



3. Performance タブに切替え、Target に要求される動作周波数やレイテンシを設定すると大凡のリソース 使用量やレイテンシが Report に表示されます。本実装では Goal を Combined とし、ターゲット周波 数を 100MHz、レイテンシを 26 cycles に設定します。

ALTERA_FP_FUNCTIONS - double_div ALTERA_FP_FUNCTIONS altera_fp_functions	
Block Diagram Functionality Performance Show signals Target Goal: Combined double_div Goal: Combined MHz clk clock conduit Conduit areset reset Performance	Target Goal : <u>Combined</u> Target : <u>100</u> MHz Target : <u>26</u> cycles
en_conduit aconduit b_conduit attera_fp_functions Hinimum acheivable latency at given frequency is 26 cycles, regist Resource Estimates: Multiplies: 28 LUTs: 2739 Memory Bits: 389120 Memory Blocks: 39	ter padding will occur to meet laten
Info: double_div: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV -fre Info: double_div: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV -fre	quency 100 -name none -noChanVal quency 100 -name none -noChanVal
	Cancel Finish

- 4. [Finish] をクリックし、Generation ダイアログ・ボックスに Generation Successful と表示されたら [Exit] をクリックして閉じます。
- 5. 生成された IP コアを Quartus Prime プロジェクトに組み込むかどうかのダイアログが表示されますので、 [Yes] をクリックして組み込みます。



4-2-4. 整数(32bit)から倍精度浮動小数への変換コアの生成

- 1. 前項の 1. と同様の操作で、 IP Catalog から ALTERA_FP_FUNCTION を開き、生成するファイル名 を "double_i2d.v"、Verilog HDL で生成します。
- 2. ALTERA_FP_FUNCTION の IP コア生成ダイアログ・ボックスが表示されます。 Function タブを下記の設 定にします。

ALTERA_FP_FUNCTIONS - double_i2d	
ALTERA_FP_FUNCTIONS altera_fp_functions	Function
Block Diagram Show signals	Family : <u>Conversions</u> Name : <u>Fixed to floating Point</u> Floating Point Data Format : <u>Double</u> Fixed Point Data Width : <u>32</u> bits Fraction : <u>0</u> bits Sign : <u>signed</u> Ports Generate an enable port : <u>ON</u>
Info: double_i2d: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV - Info: double_i2d: C:/intelfpga/17.1/quartus	-frequency 100 -name none -noChanV -frequency 100 -name none -noChanV

3. Performance タブに切替え、Target に要求される動作周波数やレイテンシを設定すると大凡のリソース 使用量やレイテンシが Report に表示されます。本実装では Goal を Combined とし、ターゲット周波 数を 100MHz、レイテンシを 3 cycles に設定します。

Block Diagram Show signals	Performance	Target : <u>100</u> MHz Target : <u>3</u> cycles
double_l2d clk_clock_conduit areset_reset_ en_conduit attera_fp_functions Goal. Target: Target: Target: Target: Minimum ach Resource Est Multiplies: LUTs: Memory Bits: Memory Bits:	imates: 0 372 0 5: 0 0 100 MHz 100 MHz 100 100 100 100 100 100 100 10	s, register padding will occur to meet laten
Info: double_i2d: C:/intelfpga/17.1/quartus//dspba/ba	ckend/windows64/cmdPolyEval -target Cyclo ckend/windows64/cmdPolyEval -target Cyclo	oneV -frequency 100 -name none -noChanV oneV -frequency 100 -name none -noChanV



- 4. [Finish]をクリックし、Generation ダイアログ・ボックスに Generation Successful と表示されたら[Exit]をクリックして閉じます。
- 5. 生成された IP コアを Quartus[®] Prime プロジェクトに組み込むかどうかのダイアログが表示されますの で、[Yes]をクリックして組み込みます。
- 4-2-5. 倍精度浮動小数から整数(32bit)への変換コアの生成
 - 1. 前項の 1. と同様の操作で、 IP Catalog から ALTERA_FP_FUNCTION を開き、生成するファイル名 を "double_d2i.v"、Verilog HDL で生成します。
 - 2. ALTERA_FP_FUNCTION の IP コア生成ダイアログ・ボックスが表示されます。 Function タブを下記の設 定にします。

ALTERA_FP_FUNCTIONS altera_fp_functions Block Diagram Characterization Block Diagram Characterization Block Diagram Characterization Function Func	Family : <u>Conversions</u> Name : <u>floating to Fixed Point</u> Floating Point Data Format : <u>Double</u> Fixed Point Data Width : <u>32</u> bits Fraction : <u>0</u> bits Sign : <u>signed</u> Ports
Info: double_d2i: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV Info: double_d2i: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV Info: double_d2i: C:/intelfpga/17.1/quartus//dspba/backend/windows64/cmdPolyEval -target CycloneV	Generate an enable port : <u>ON</u>



3. Performance タブに切替え、Target に要求される動作周波数やレイテンシを設定すると大凡のリソース 使用量やレイテンシが Report に表示されます。本実装では Goal を Combined とし、ターゲット周波 数を 100MHz、レイテンシを 3 cycles に設定します。

	Functionality	Performance	Target : <u>100</u> MHz Target : <u>3</u> cycles
cik clock conduit q	Target: Target: Target :	100 MHz 3 cycles	
a conduit altera_fp_functions	Minimum acheiv Resource Estim Multiplies:	vable latency at given frequency is 3 cyc nates: 0	des, register padding will occur to meel
	Memory Bits: Memory Bits:	355 0 0	
	<pre></pre>		
) Info: double_d2i: C:/intelfpga/17.1 Info: double_d2i: C:/intelfpga/17.1	/quartus//dspba/back /quartus//dspba/back	end/windows64/cmdPolyEval -target C end/windows64/cmdPolyEval -target C	ycloneV -frequency 100 -name none -nod ycloneV -frequency 100 -name none -nod

- 4. [Finish] をクリックし、Generation ダイアログ・ボックスに Generation Successful と表示されたら [Exit] をクリックして閉じます。
- 5. 生成された IP コアを Quartus Prime プロジェクトに組み込むかどうかのダイアログが表示されますので、 [Yes] をクリックして組み込みます。

4-2-6. 倍精度浮動小数比較コアの生成

 Quartus Prime の IP Catalog ウィンドウの Installed IP ⇒ Library ⇒ Basic Function ⇒ Arithmetic ⇒ ALTFP_COMPARE を選択します。ダブルクリックか [Add] をクリックし、IP variation file name に生成す るファイルパスおよびファイル名、IP variation file type に HDL タイプを選択し [OK] をクリックします。 本実装では、生成するファイル名を "double_cmp.v"、Verilog HDL で生成します。



2. ALTFP_COMPARE の IP コア生成ダイアログ・ボックスが表示されます。 General タブを下記の設定にします。

🔌 MegaWizard Plug-In Manager [page	1 of 4]		? ×	
ALTFP_COMPAR	ŧE	About Documer	ntation	
Parameter 2 EDA 3 Summary Settings General 2 General General 2 Image: Control of C	Currently selected device family: What is the floating point format? Single precision (32 bits) Couble precision (32 bits) Single extended precision (43 bits to 64 bits) How wide should the 'dataa' input, and 'datab' input bus How wide should the exponent field be? Mantissa width = (data input width) - (exponent field width) - 1 What is the output latency in clock cycles?	: Cydone V Match project/de ses be? 64 11 52 1	What What cycle	t is the floating point format? <u>Double precision (64 bits)</u> t is the output latency in clock s? <u>1</u> Cycles
70 ht i 1 cos				



3. General2 タブに切替え、下記の設定にし、[Finish] をクリックします。

S MegaWizard Plug-In Manager [page 2 of 4]	<u>?</u> ×
ALTFP_COMPARE	About Documentation
Parameter ZEDA Summary General General	
denteral Z double_cmp datab(63.0) aeb agb datab(63.0) aeb agb ab clock ageb clock cycles: 1 Double Precision Exponent Width: 11 Mantissa Width: 52 Mantissa Width: 52 Or you want to create optional inputs? If Create a dock enable port If Create a dock enable port	Which output do you want? $a = b : ON$ $a <> b : ON$ $a <> b : ON$ $a > b : ON$ $a > b : ON$ $a < contract of the two states of the two states of the two states of two s$
Resource Usage 79 lut + 6 reg Cancel	Create a clock enable port : <u>ON</u>

4. Summary タブが表示されたら、再度 [Finish] をクリックします。

MegaWizard Plug-In ALTFP Parameter Settings 2 EDA	n Manager [p COMP 	age 4 of 4] ARE	About Documentation
double_cn dataa[63.0] datab[63.0] clock clock	np aeb aneb agb ageb alb	Turn on the files you wish automatically generated, generate the selected file MegaWizard Plug-In Mana The MegaWizard Plug-In N C:\WORK\20180619_DOC File	to generate. A gray checkmark indicates a file that is and a green checkmark indicates an optional file. Click Finish t s. The state of each checkbox is maintained in subsequent ger sessions. Anaager creates the selected files in the following directory: :\ci_double_fp\ Description
actr Clock Cycles: 1 Double Precision Exponent Width: 11 Mantissa Width: 52	aleb ->	double_cmp.v double_cmp.inc double_cmp.cmp double_cmp.bsf double_cmp_inst.v double_cmp_bv.v	Variation file AHDL Include file VHDL component declaration file Quartus Prime symbol file Instantiation template file Verilog HDL black-box file
Resource Usage 79 lut + 6 reg			Cancel < Back Next >

5. 生成された IP コアを Quartus Prime プロジェクトに組み込むかどうかのダイアログが表示されますので、 [Yes] をクリックして組み込みます。

4-3. カスタム・インストラクションのトップ・レベル・ファイル

本実装で使用するカスタム・インストラクション・モジュールのトップ・レベル HDL を下記に示します。





<pre>assign done = (n == `INST_WRX)? start : (n == `INST_WRY)? start : (n == `INST_RDL)? start : (n == `INST_RDH)? start : (n == `INST_RDY)? start : (n == `INST_ADD)? ((count == 0)? 1 : 0) : (n == `INST_SUB? ((count == 0)? 1 : 0) : (n == `INST_DIV)? ((count == 0)? 1 : 0) : (n == `INST_DIV)? ((count == 0)? 1 : 0) : (n == `INST_DI?? ((count == 0)? 1 : 0) : (n == `INST_DI?? ((count == 0)? 1 : 0) : (n == `INST_DI?? ((count == 0)? 1 : 0) : (n == `INST_DI?? ((count == 0)? 1 : 0) : (n == `INST_LE)? ((count == 0)? 1 : 0) : (n == `INST_GT)? ((count == 0)? 1 : 0) : (n == `INST_GE)? ((count == 0)? 1 : 0) : (n == `INST_GE)? ((count == 0)? 1 : 0) : (n == `INST_NE)? ((count == 0)? 1 : 0) : (n == `INST_NE)? ((count == 0)? 1 : 0) : 0;</pre>	WRX,WRY,RDL,RDH,RDY は start を done に引き回し、その他は レイテンシ・カウンタが 0 で done をアサート
<pre>assign result = (n == `INST_WRX)? 0 : (n == `INST_MRV)? 0 : (n == `INST_RDL)? datax[31: 0] : (n == `INST_RDH)? datax[63:32] : (n == `INST_RDY)? datax[31: 0] : (n == `INST_ADD)? add_result[63:32] : (n == `INST_STUB? add_result[63:32] : (n == `INST_DIV)? div_result[63:32] : (n == `INST_DIV)? div_result[63:32] : (n == `INST_DIV)? div_result[63:32] : (n == `INST_DIV]? div_result[63:32] : (n == `INST_DIV]? div_result[63:32] : (n == `INST_LT)? {30'h0, alt} : (n == `INST_LE)? {30'h0, agt} : (n == `INST_GE)? {30'h0, agg} : (n == `INST_CA)? {30'h0, aeg} : (n == `INST_NE)? {30'h0, ane} : 0; endmodule</pre>	WRX,WRY は書込みのため result は 0、RDL,RDH,RDY は 演算結果レジスタの値、演算は 演算コアの出力、比較は LSB に比較結果



4-4. Component Editor によるカスタム・インストラクションの作成

Platform Designer にカスタム・インストラクション・モジュールを登録します。

1. New Component メニューから Component Editor を起動して、 Component Type タブで Name および Display name 、Group を設定します。

👗 Component	Editor - double_fp_hw.tcl*	I.V.
<u>File</u> <u>T</u> emplates	<u>B</u> eta <u>V</u> iew	Name : <u>double_fp</u>
Component Typ About Comp	pe 😸 Block Symbol 🖄 Files 🔅 Parameters 🔅 Signals & Interfaces ponent Type	Display name : <u>Double precision floating point</u>
		four arithmetic operation
Name: Display name:	double_fp Double percision floating point four arithmetic operation	Group : <u>Custom Instruction Modules</u>
Version:	1.0	
Group:	Custom Instruction Modules	
Description:		
Icon:		
Documentation:	Title URL	
	+ -	

2. Files タブで Synthesis Files および Verilog Simulation Files を設定します。

ionent Tvoe 💠 🛛 Block Svm	ibol 🖾 Files 😹 Parameters 🏻	Signals & Interfaces	214 C
bout Files			
nesis Files files describe this component	's implementation, and will be created wh	en a Quartus synthesis model is ge	enerated.
Output Path	Source File	Type	Attributes
double fpu.v	double fpu.v	Verilog HDL	Top-level File
double_add.v	double_add.v	Verilog HDL	no attributes
double_mul.v	double_mul.v	Verilog HDL	no attributes
double div.v	double div.v	Verilog HDL	no attributes
double_i2d.v	double_i2d.v	Verilog HDL	no attributes
JEIJ9:	1. 1	Manile - UDI	
Add File Remove File	Analyze Synthesis Files Create Sy	nthesis File from Signals	In arrestorer
Add File Remove File evel Module: (Analyze files to og Simulation Files files will be produced when a	Verilog simulation model is generated.	nthesis File from Signals	Attributes
Add File Remove File avel Module: (Analyze files to be Simulation Files files will be produced when a Output Path double from:	Verilog simulation model is generated.	Type	Attributes
Add File. Remove File evel Module: (Analyze files to be Simulation Files files will be produced when a Output Path double_fpu.v double_dd un	Verilog simulation model is generated.	nthesis File from Signels Type Verilog HDL Verilog HDL	Attributes
Add File Remove File avel Module: (Analyze files to be Simulation Files files will be produced when a Output Path double_fpu.x double_add.yo double_add.yo	Analyze Synthesis Files Oreste Sy select module) Verilog simulation model is generated. Source File double_add sim/double_add/vo double_mul sim/double_add/vo	nthesis File from Signals Type Verilog HDL Verilog HDL Verilog HDL	Attributes no attributes no attributes
Add File Remove File avel Module: (Analyze files to be Simulation Files files will be produced when a Output Path double_fpu.v double_mul.vo double_mul.vo double_mul.vo	Verilog simulation model is generated. Verilog simulation model is generated. Verilog simulation model is generated. Source File double_fpuv double_and_sim/double_add.vo double_mulvo double_div sim/double_mulvo	Type Verilog HDL Verilog HDL Verilog HDL Verilog HDL	Attributes no attributes no attributes no attributes

[Add File...] をクリックしてファイルを登録します。トップ・レベル・ファイルは "double_fpu.v" を追加し、それ以 外は 3-2 章で生成したファイルを追加して行きます。

シミュレーション用ファイルは、トップ・レベル・ファイル(double_fpu.v)と比較演算コア (double_cmp.v) は合成 用ファイルと同じものを使用しますが、 ALTERA_FP_FUNCTION で生成したコアは .vo ファイル(Verilog-HDL ネットリスト・ファイル)が生成されますので、そちらを使用します。

.vo ファイルは、<*生成したコア名*>_sim フォルダに生成されていますので、こちらを追加してください。



なお、double_div_sim フォルダには、.hex(メモリ初期化ファイル)も生成されますので、こちらも追加してください。

全てのファイルを登録し終わったら、各ファイルの Type を確認します。

.v または .vo ファイルの Type は "Verilog HDL" とし、 .hex ファイルの Type は "HEX Data" としてください。

合成用ファイルの登録結果

Output Path	Source File	Туре	Attributes	
double_fpu.v	double_fpu.v	Verilog HDL	Top-level File	
double_add.v	double_add.v	Verilog HDL	no attributes	
double_mul.v	double_mul.v	Verilog HDL	no attributes	
double_div.v	double_div.v	Verilog HDL	no attributes	
double_i2d.v	double_i2d.v	Verilog HDL	no attributes	
double_d2i.v	double_d2i.v	Verilog HDL	no attributes	
double_cmp.v	double_cmp.v	Verilog HDL	no attributes	

シミュレーション用ファイルの登録結果

Output Path	Source File	Туре	Attributes
double_fpu.v	double_fpu.v	Verilog HDL	no attributes
double_add.vo	double_add_sim/double_add.vo	Verilog HDL	no attributes
double_mul.vo	double_mul_sim/double_mul.vo	Verilog HDL	no attributes
double_div.vo	double_div_sim/double_div.vo	Verilog HDL	no attributes
double_div_memoryC0_uid146_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid147_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid148_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double div memoryC0 uid149 invTables lutmem.hex	double div sim/double div m	HEX Data	no attributes
double_div_memoryC0_uid150_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid151_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid152_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid153_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid154_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid155_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC0_uid156_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double div memoryC0 uid157 invTables lutmem.hex	double div sim/double div m	HEX Data	no attributes
double_div_memoryC1_uid161_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid162_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid163_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid164_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid165_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid1bb_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid167_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid1b8_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid1b9_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC1_uid170_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC2_uid173_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC2_uid174_inv1ables_lutmem.nex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC2_uid175_inv1ables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryO2_uid176_inv1ables_lutmem.nex	double_div_sim/double_div_m		no attributes
double_div_memoryO2_uid177_inv1ables_lutmem.nex	double_div_sim/double_div_m		no attributes
double_div_memoryC2_uid178_inv1ables_lutmem.nex	double_div_sim/double_div_m		no attributes
double_div_memoryC2_uid179_invTables_lutmem.nex	double_div_sim/double_div_m		no attributes
double_uiv_memoryO2_uiu160_invTables_lutmem.hex	double_div_sim/double_div_m		no attributes
double_div_memoryC3_uid103_invTables_lutmembex	double_div_sim/double_div_m		no attributes
double_div_memoryC3_uid185_invTables_lutmem.hex	double_div_sim/double_div_m	HEX Data	no attributes
double_uiv_memoryC3_uiu185_invTables_lutmem.hex	double_div_sim/double_div_m		no attributes
double_div_memoryC3_uid100_invTables_lutmembex	double_div_sim/double_div_m	HEX Data	no attributes
double_div_memoryC3_uid188_invTables_lutmem.hex	double_div_sim/double_div_m	HEV Data	no attributes
double_div_memoryC4_uid101_invTables_lutmem.hev	double div sim/double div m	HEX Data	no attributes
double_div_memoryC4_uid102_invTables_lutmem.hev	double div sim/double div m	HEX Data	no attributes
double_div_memoryC4_uid103_invTables_lutmem.hev	double_div_sim/double_div_m	HEX Data	no attributes
double_ind_wo	double i2d sim/double i2duo	Verilog HDI	no attributes
double_120.vo	double_120_stm/double_120.v0	Verilog HDI	no attributes
double_cmp.v	double_cmp.v	Verilog HDI	no attributes
	CONTRACTOR CONTRACTOR OF CONTRACTOR		THE REAL PROPERTY OF THE REAL



3. Signal & Interface タブで <<add interface>> をクリックして Custom Instruction Slave を選択します。

Component Editor - double_fp_hw.tcl*			×
File Templates Beta View			
Component Type 🙁 Block Symbol 😣 Files	⊠ Parameters	🛿 🛛 Signals & Interfaces	× _ d =
▶ About Signals			
Name			
Kadd interface	I.		
AHB Master			
AHB Slave			
APB Master			
APB Slave			
AXI 4 Stream Master			
AXI 4 Stream Slave			
AXI Master			
AXI Slave			
AXI4 Master			
AXI4 Slave			
AXI4Lite Master			
AXI4Lite Slave			
Avalon Memory Mapped Master			
Avalon Memory Mapped Slave			
Avalon Streaming Sink			
Avalon Streaming Source			
Clock Input			
Clock Output			
Conduit			
Custom Instruction Master			
Custom Instruction Slave			
HSSI Bonded Clock Input			
HSSI Bonded Clock Output			
HSSI Serial Clock Input			

4. 作成された nios_custom_instruction_slave の下の <<add signal>> をクリックして clk を選択します。

👶 Component Editor	- double_fp_hw.tcl*				×
File Templates Beta	View				
Component Type 🛛	Block Symbol 🛛	Files 🖾	Parameters 🖾	Signals & Interfaces	× _ d 🗆
▹ About Signals					
	Name				
► nios_custom_ins	truction_slave Custo	m In			
Kadd a					
b					
clk					
cik_en					
dataa datab					
done					
n					
readra					
reset					
reset_	req				
result					
start					
writer	2				

5. 同様に <<add signal>> をクリックして clk_en、dataa、datab、done、n、reset、result、start の信号を生成 します。

* Component Editor - double_fp_hw.tcl*	×
<u>File Templates B</u> eta <u>V</u> iew	
Component Type 🛛 Block Symbol 🔅 Files 🌣 Parameters 🖄 Signals & Interfaces 🖄 🗕 🗖	f 🗆
About Signals	
Name	
 nios_custom_instruction_slave_Custom Instruction Sia nios_custom_instruction_slave_cik [1] cik nios_custom_instruction_slave_cik en [1] cik_en nios_custom_instruction_slave_data [32] datas nios_custom_instruction_slave_data [32] datab nios_custom_instruction_slave_done [1] done nios_custom_instruction_slave_n [2] n nios_custom_instruction_slave_reset [1] reset nios_custom_instruction_slave_reset[32] result 	
D- nios_custom_instruction_slave_start [1] start < <add signal="">> <<add interface="">></add></add>	



 各信号の設定値を変更します。Name は最初の "nios_custom_instruction_slave_" を削除して、n は Width を"5"に変更してください。それ以外はデフォルトのままで問題無いと思いますが、念のため下記 の設定になっているか確認してください。

Name	Signal Type	Width	Direction				
clk	clk	1	input				
clk_en	clk_en	32	input				
dataa	dataa	32	input				
datab	datab	1	input				
done	done	1	output				
n	n	5	input				
reset	reset	1	input				
result	result	32	output				
start	start	1	input				

設定完了後は下記のようになります。

Component Editor -	double_fp_hw.tcl*				×
<u>File lemplates B</u> eta	View				
Component Type 🛛	Block Symbol 🛛 🕅	Files 🛛	Parameters 🖾	Signals & Interfaces 🛛 🕄	- ದೆ 🗆
▹ About Signals					
	Name				
nios_custom_inst	ruction_slave Cust in ia ib	om Instructio	n Sla		

7. [Finish] をクリックし、Save Changes メッセージ・ボックスが表示されたら [Yes, Save] をクリックします。

Save C	hanges
?	You are about to save Double precision floating point four arithmetic operation 1.0 to:
	The following files will be written into your component's folder: double_fp_hw.tcl (component description)
	Do you wish to save changes before exiting?
	Yes, Save No, Don't Save Cancel

8. Save and Refresh System Completed ダイアログ・ボックスが表示され、 completed successfully が表示さ れたら成功です。[Close] をクリックしてウィンドウを閉じます。





9. Platform Designer の IP Catalog に、追加されたモジュールが表示されますので、このモジュールをダブ ルクリックまたは [Add] をクリックして System Contents に追加します。



10. Nios II の custom_instraction_master と、追加したモジュールの nios_custom_instraction_slave を接続し、 Base 列の Opecode を "<u>224</u>" に変更します。

+ s	öystem (≍ ▲	Contents 🕺 Address	s Map	ents 🖾				
+	Use	Connections	Name	Description	Export	Clock	Base	End
 Section 2 Section	G		□ clk_0 clk_in clk_in_reset clk clk reset	Clock Source Clock Input Reset Input Clock Output Reset Output	clk reset Double-	exported		
* ¥	R		the price of	Nios II Processor Clock Input Reset Input Avalon Memory Mapped M Avalon Memory Mapped M Interrupt Receiver Reset Output Avalon Memory Mapped SI	Double- Double- Double- Double- Double- Double- Double-	cik_0 [cik] [cik] [cik] [cik] [cik] [cik]	IRQ 0 ■ 0×0002_0000	IRQ 31 0×0002_07ff
	1		custom_instruction_master	Custom Instruction Master	Double	4		
	दा दा दा दा दा दा		 ➡ onchip_memory2_0 ➡ led_pio ➡ dipsw_pio ➡ button_pio ➡ jtag_uart_0 ➡ sysid_qsys_0 ➡ double_fp_0 	On-Chip Memory (RAM or PIO (Parallel I/O) PIO (Parallel I/O) PIO (Parallel I/O) JTAG UART System ID Peripheral Double precision floating p Output Reference St		clk_0 clk_0 clk_0 clk_0 clk_0 clk_0	0x0000_0000 0x0002_0800 0x0002_0800 0x0002_0800 0x0002_0800 0x0002_0800 0x0002_0800 0x0002_0800	0x0001_ffff 0x0002_081f 0x0002_090f 0x0002_0&0f 0x0002_0b07 0x0002_0c07

11. Platform Designer を Generate し、Quartus Prime でコンパイルします。



5. <u>検証</u>

5-1. ModelSim-Intel FPGA Edition 10.5b (以降、ModelSim)によるシミュレーション

Nios II Software Build Tools for Eclipse (以降、Nios II SBT) と ModelSim を使用して、カスタム・インストラクション・ モジュールが正常に動作するか確認します。Nios II SBT を起動して新規にプロジェクトを作成し、下記の設定 を行います。

- BSP Editor を起動して、enable_sim_optimize にチェックを入れます。
- 2.5 章の通り、Makefile にコンパイル・オプションを設定します。
- オプティマイズ・オプションは -O0(off) に設定します。

5-1-1.四則演算の確認

1. Nios II SBT のプロジェクトの main() 関数を下記のとおり記述します。

```
* main.c[]
#include <stdio.h>
#include <system.h>
int main(void)
{
    double A = 1.2345678;
    double B = 2.3456789;
    volatile double R;
    R = A + B;
    R = A + B;
    R = A - B;
    R = A * B;
    R = A / B;
    return 0;
}
```

正常にビルドできることを確認した後、Project Explorer のアプリケーション・プロジェクトを選択し、右クリックで Run As ⇒ 4 Nios II ModelSim を起動します。





3. ModelSim でシミュレーションが実行されます。この時、Nios II からカスタム・インストラクション・モジュー ルに接続されたノードを観測します。

dauble for					
— double_rpu —	1000			王	同人たち
🤣 dk	St1				
i reset	St0				الما الالمام القاقات المحمد الدمع المحمد
🧔 dk_en	St0				
🗉 🍲 n	16		000000000000000000000000000000000000000	())))))))))))))))))))))))))))))))))))))	
🗉 🍲 dataa	00000000	()))))))))))))))))))))))))))))))))))))	(1))())) 4d935689 (1) (1))())))))))	()))))))))))))))))))))))))))))))))))))	()()) () ()()()()()()()()()()()()()()()
🖅 🏑 datab	ffff9590	1) (4002c3f3))) ()))))))))))))))))))))))))))))))	(1)))))) 4002c3f3 (0 ())))))))))))))))	()) ()) ()) () () () () () () () () () (()))))))))))))))))))))))))))))))))))))
💶 📥 result	00000000	00000 XXX X X X 00000000	(400ca)))))))))))))))))))))))))))))))))		17ff00000 () () () ()
📣 start	St0				
📥 done	St0				
	15	15 11111111111	1000000015	7777777 15	111111111111111111111111111111111111111
	0.526316	1 23457 1 3 58025	1 23457 1-1 11111	1 23457 1 289590	1 23457 10 526316
- double add -					
n .() a	0 526316	1 23457 X 3 59025	1 23457	1 23457 12 80500	1 23457 Y 0 526316
h h	NAN				1223457 (0.326510 11111111111111111111111111111111111
an an	Sto			<u></u>	
	510	╏╴╴╴╴╴╴╴╴╴╴╴╴			
opsei	3.0				
te="top q	-1.11111	+0 <u>) // / +0 / 3.58025</u>	X-1.11111		
+	0.526316	1.2345/ 1.3.58025	1.23457 1-1.11111		1.23457 10.526316
1	NAN		<u>41 11 11 2.34568 111 111 111 111</u>		111111111111111111111111111111111111111
🗄 🚓 en	St0				
🗄 🛟 d	2.89590	+0) 2.89590	
double_div					
🗄 🤣 a	0.526316	(1.23457) 3.58025	(1.23457 (-1.11111	(1.23457 (2.89590	(1.23457 (0.526316
😐 🎝 b	NAN	1 01 2.34568 00 000000000000000000000000000000000	01())))(2.34568))()(01))))()))))))	000000000000000000000000000000000000000	000000000000000000000000000000000000000
🖪 👍 en	St0				
🖬 📥 q	0.526316	+0			(ONAN 0.526316

加算の詳細

— double_fpu —						
👍 dk	St1					
👍 reset	St0	والاحتدار والمتحار والمتحار والمتحا				
📣 dk_en	StO					
🗉 🎝 n	16	χο	.5		الكر المسالح	<u>(0 (8)</u> 4
🖅 🎝 dataa	00000000	00000000 (2a5b 1d5d) (0000 (0000	. 4c935689			(0000 (0001 (0000000
🖅 🎝 datab	ffff9590	00000000 (3ff3c0ca) 00000000	4002c3f3			(0000) (ffffffe8 (0000000)
🖅 🕂 🕞 🕞	00000000	00000000) 7f20) 7c90) 00000000) 400ca	458 (0000000 (61c0e538
📣 start	St0		(3)		i 🔶	
🔩 done	St0					
😐 - 🧇 count	15	15	χ8 χ7	(6) (5) (4) (3	<u>)2)1)0</u>	31 (15
😐 🤣 datax	0.526316	+0 (1.23457		النائذ النائة فتخذ		3.58025
double_add						5)7
🖽 🛷 a	0.526316	+0 1.23457				8.58025
😐 🛷 b	NAN	+0 (1.23457 (+0 (5.79	2.34568			(5.79 (NAN) +0
🖽 🤣 en	St0		+		(4)	
🕳 🍲 opSel	St0				¥	
🕒 🛟 q	-1.11111	+0		<u>(2.19 (9.97) +0</u>	3.580	25

- ① WRX(#0) が実行され、datax に 64bit データをロード
- ② ADD(#5) が実行され、被加数 64bit が確定。 opSel は Hi
- ③ count が 8 で初期化され、カウントダウン開始
- ④ double_add の result が確定し、演算結果の上位 32bit を出力。カウントが0になり ADD(#5)
 の done をアサートして ADD(#5) の実行を終了
- ⑤ 演算結果を datax に格納
- ⑥ RDY(#4) が実行され、演算結果 (下位 32bit) を出力



減算の詳細

— double_fpu —	a second		التباري البري البري البري	
👍 dk	St1			
📣 reset	St0	بالمتكالميك بمنكر لمتكر المتكر المتكر بجلال الجبري كمكر كمك كمك المتكر كردي كرا	يحديها لمقط فمعط كوحد بمدعر	
📣 dk_en	St0			
🗉 🌧 n	16	χε χο χε χο χε χο χε) (0 (28) (4	
🗉 🎝 dataa	00000000) (0001) (2a5b1d5d) (4002) (2a5b) (4002) (2a5b) 4<:935689	(2a5b) 4c93) (000	00000
🖅 🎝 datab	ffff9590	χοοοφ J χ3ff3coca χοοοφ χοοοφ χοοοφ χοοοφ μ4οοχc3f3	χοοοφοσοο	
🖅 🕂 🔁 result	00000000	00000000 1 400ta458)bff1c71c)0000000)6ec	b8fb5
👍 start	St0			
📥 done	St0			
	15		(0)31 (15	(6)
🖅 🕂 🕂 🛨	0.526316	3.58025 X1.23457	-1.11111	V
double_add			57	
🖪 🕁 a	0.526316	3.58025 1.23457	-1.11111	
😟 🕁 b	NAN) 6.96) 1.23457) 6.96) 3.51) 6.96) 3.51 2.34568	(3.51)(6.34)+0	
💽 🎝 en	St0		(4)	
🖅 🤣 opSel	St0			
🖽 📥 q	-1.11111	3.58025)=1.11111	

- ① WRX(#0) が実行され、datax に 64bit データをロード
- ② SUB(#6) が実行され、被乗数 64bit が確定。 opSel は Lo
- ③ count が 8 で初期化され、カウントダウン開始
- ④ double_add の result が確定し、演算結果の上位 32bit を出力。カウントが 0 になり SUB(#6) の done をアサートして SUB(#6) の実行を終了
- ⑤ 演算結果を datax に格納
- ⑥ RDY(#4) が実行され、演算結果 (下位 32bit) を出力

乗算の詳細									
double_fpu	St1								
👍 reset 📣 dk en	St0 St0			;;					
	16 00000000	0 400 ¥ 2a561d5d	1 4002c3f3	1 4-985689				<u>)0</u>)4	3 11 00000000
-4 datab	ffff9590	400) 3ff3c0ca	(4002c3f3	1,1030003			1 400 725cd)8c96)000	0) 00000000
start	St0		D	i	3		A 400728C0	χοσομ χαει	
count	15	15		2 14	χ3	<u>)2)1</u>	<u>χο χ</u> 31	<u>) 15</u>	6
double_mul	0.526316	-1.11111 (1.234	57				5.	90	
∎-4, b	0.526316 NAN	<u>-1.11111) 1.234</u> 2.34) 1.23457	2.34568	(2.34568			(4)	90) -4.93) 9.1	2) +0
∎-4> en ∎-4> q	St0 2.89590	+0					<u>) 2.89590</u>		

- ① WRX(#0) が実行され、datax に 64bit データをロード
- ② MUL(#7) が実行され、被乗数 64bit が確定
- ③ count が 4 で初期化され、カウントダウン開始
- ④ double_mul の result が確定し、演算結果の上位 32bit を出力。カウントが 0 になり MUL(#7) の done をアサートして MUL(#7) の実行を終了
- ⑤ 演算結果を datax に格納
- ⑥ RDY(#4) が実行され、演算結果 (下位 32bit) を出力



除算の詳細

— double_fpu —			
👍 dk	St1		
👍 reset	St0		
📣 dk_en	St0		1
🖽 👍 n	16	24 X0 X24 X0 X8 X4	
🖅 🎝 dataa	00000000	000 (2a5b1d5d) (0000 (0000) (4c955689) (0000) (000000) (000000) (000000) (000000) (000000) (000000) (000000) (000000) (00000000) (0000000) (0000000) (000000000) (000000000) (0000000000	
🖅 🎝 datab	ffff9590	000 () 3ff3c0ca () 0000 () 0000) 4002c3f3	
🖅 🚓 result	00000000	00000000 X X1Fedd794 X00000000 X X2c42011	8
istart 🍫	St0		
👍 done	St0		
	15	15 (25) (25) (24) (23) (3) (2) (1) (0) (31) (15)	
庄 🤣 datax	0.526316	2.89590 (1.23457 (2)	5)
double_div			~
🖽 🎝 a	0.526316	2.89590 (1.23457 0.526316 0.526316	
🖽 🤣 b	NAN	5.0 X 1.23457 X 5.09 X 5.79 X 2.34568 4.1 X 5.79 X +0 X 5.79 X +0 X 5.79 X +0	
🗉 🎝 en	St0		
🛨 🕁 q	0.526316	+0 \(\).526316	

- ① WRX(#0) が実行され、datax に 64bit データをロード
- ② DIV(#8) が実行され、被除数 64bit が確定
- ③ count が 25 で初期化され、カウントダウン開始
- ④ double_div の result が確定し、演算結果の上位 32bit を出力。カウントが 0 になり DIV(#8)
 の done をアサートして DIV(#8) の実行を終了
- ⑤ 演算結果を datax に格納
- ⑥ RDY(#4) が実行され、演算結果 (下位 32bit) を出力



5-1-2. 型変換の確認

1. Nios II SBT のプロジェクトの main() 関数を下記のとおり記述します。

```
* main.c[]
#include <stdio.h>
#include <system.h>
int main(void)
{
    int A = 123456;
    volatile double R;
    R = (double)A;
    A = (int)R;
    return 0;
}
```

- 2. 正常にビルドできることを確認した後、 Project Explorer のアプリケーション・プロジェクトを選択し、右クリ ックで Run As ⇒ 4 Nios II ModelSim を起動します。
- 3. ModelSim でシミュレーションを実行します。この時、Nios II からカスタム・インストラクション・モジュール に接続されたノードを観測します。

- double fou -												
doddicpd	St1	inn	整数;	から倍	精度	£ IUI	w	… 倍	精度浮	動小	数	nn
set	StO											
_ A dk_en	Stu		NOT EL	In Stellar v					- 市ケ米ト			vv
+	10		注到	IV£X ∖	1.14			<u>+ 72</u>	り金剱	1,10	⊨÷i	
🛨 🎝 dataa	00000000		10001e240	10001#e4	10	0001ffe4	<u>4 μ. </u>		<u> </u>	0000000	<u>/</u>	
🛨 🛫 datab	11119590		100000000		1 10			للللل	00000 <u>)</u>)) 40fe2400)000
🛨 🔶 result	00000000	0000	11114.	. χοσοσοσο						λ	10	100000
istart 🌮	St0											
👍 done	St0											
💽 🌙 count	15	15		<u>) 15</u>								5
💽 🌙 datax	6.09954e	+0)	123456.							6.09	954e-3
double_d2i			-									
😐 🌙 a	NAN		(6.09954e		((+O					0 123456.	(5	6
👍 areset	St0											
👍 dk	St1	INA	innnni	Innnn	nnn	ndor	nnn	INNNN	innnn	INNNN	INN	nnı
🚛 🌙 en	St0										n i	
	123456	0								Y Y	123456	6
- double i2d												
	0	YYY	Y 123456) 131044	Ĭn	131044	Y Y 1	TTTTT	YYYY		Y	Ĭ1.
areset	SHO		1	h 1010 11			AA					
dk dk	St1	inn	10000	nnnnr	000	hdor	nnn	0000	Innnn	00000	nn.	nnr
n den	SHO											
	172456	2 147		AFC								



整数から倍精度浮動小数への型変換の詳細



- ① I2D(#9) が実行され、data の整数 (32bit) が確定
- ② count が2で初期化され、カウントダウン開始
- ③ double_i2d の result が確定し、演算結果の上位 32bit を出力。カウントが 0 になり I2D(#9)
 の done をアサートして I2D(#9) の実行を終了
- ④ 演算結果を datax に格納
- ⑤ RDY(#4) が実行され、演算結果 (下位 32bit) を出力

倍精度浮動小数から整数への型変換の詳細



- ① D2I(#10) が実行され、dataa , datab を連結して倍精度浮動小数(64bit)が確定
- ② count が2で初期化され、カウントダウン開始
- ③ double_d2i の result が確定し、演算結果を出力。カウントが 0 になり D2I(#10) の done をア サートして D2I(#10) の実行を終了



5-1-3. 比較演算の確認

1. Nios II SBT のプロジェクトの main() 関数を下記のとおり記述します。

```
* main.c[]
#include <stdio.h>
#include <system.h>
int main(void)
{
    double A = 1.2345678;
    double B = 2.3456789;
    volatile int C = 0;
    if(A < B) C++;
    if(A <= B) C++;
    if(A > B) C++;
    if(A >= B) C++;
    if(A == B) C++;
    if(A != B) C++;
    if(A != B) C++;
    if(A != B) C++;
    if(A != B) C++;
    return C;
}
```

- 2. 正常にビルドできることを確認した後、Project Explorer のアプリケーション・プロジェクトを選択し、右クリ ックで Run As ⇒ 4 Nios II ModelSim を起動します。
- 3. ModelSim でシミュレーションを実行します。この時、Nios II からカスタム・インストラクション・モジュール に接続されたノードを観測します。

— d								r==					1					Ē					(F==)				Ē		
4	e dk	St1	UDAUDID	.0.00		ກວ່າແບບການ	ພາກການການ		n D-IUI			utmuum	າດແບບທ	D-10001	U U BU		ntonom		unimuu		2001000000	mmuunn	1 KUUU	man	ULTER I	in nauranda			
4	reset	St0	A K		в			IA	$\leq =$	В				A >	ЪB				\rightarrow	÷ 1	5		IA I		в		A		В
4	dk_en	St0						1 1											ш'n				l n _ r				- I	LП	-
04	n	16	101 (31)	000	mimmo	100 00 0	0 11104	10 1	1	011 110	10114-1	ntronter	TOTIOT	10 1	111,000	nonor	mmm	1001100	1111	omonic	manaa	monom	IO ION	11010		100 11	1 1000	0.1011	10100
n-4	dataa	000000000		10				1.0)010)0	-	1001100111	enternom	0000000					000000000			10,000	1001		1001001		10000	1011001	10-10		DODEOR
04	datab	ffff9590	mont	101	minim	110100100	1001-0110	0	01.001	10011111		atroara	TOTIOT	n ona	10	momon	nomme	tiom ot		omonic	manaa	momonix	10/10/1	101	1000	$m \rightarrow n$	nama n	0.1011	1001000
n 4		00000001	0000000		00000000	0	000	100 1	Y co.	. 111 100	MI0	111 000000	000		100000	000 (1)	NUMBER	0000	MILY.	Old geoge	1 000) CODOOD	0 OY	10 oc	0000000	10 11	mor) (()] coo	0000
- 1	start	St0	0 0					0 0						h n	1								n n						1
4		SIO	0	h L				n i	'n 📕					h n					n l n				In I					n n	1
n -4	count	15	15	П Т					11115))	11:5					(<u>5</u>				OX 5				10T	15
04	datax	+0	0 112.	14.9	1)66e-324			11.2	0141-540	6e-324				11.2	+0				11.2	+0			11.2	14,9406	6e-324		- 1	11.21	6
— d																													
4	adr	St0						i										i i					i				i		
4	dk en	SIO	L [1															l í						
4	dock	St1	Doctorio	n n	nutionan	ກຕຳກາດກາງ	innonnini	namen	11111	ndumnum	nafa i nana n	infone Loopi	ndennand	Information	10 0000	u formore	ndentetern	ninnnin	nnimni	minim	norrafinanan	motoromo	Home Property	nn or	mairm	mandr	unun 🖣	inmunni	10101001
n-4	dataa	+0	0 112	149	0668-324			11.2	0140940	66-374				112	+0				1121	+0			11.2	140406	66-74		-	11.2)	0
0.4	datab	NAN	101-0000	100	000000	000000	tionronte	00000		100100000	aicman	ntrontor	TOTTOT	NT OUTC	10100100	momon	robroramia	tomor	DOTIN	omonic	mainaa	ατόποπα	DOTON	YND	notion	000001	nonne o	0700	000000
	aeb	St0																			001000000								
- 4	agb	St0																		i l									
4	ageb	SHO																											
4	ab	S*1																											
4	aleb	St1		-											-					-				- 1					-
	aneh	SP1						-							-														-

A < B 比較の詳細



- ① WRX(#0) が実行され、 datax に 64bit データをロード
- CMP_LT(#11) が実行され、比較数 64bit が確定
- ③ count が0で初期化され、カウントダウン開始
- ④ double_cmp の比較結果が確定
- ⑤ alb を result の LSB に出力し、CMP_LT(#11) の done をアサート

その他の比較演算の詳細

基本動作は同様で、オペコードに合せて double_cmp からの出力を選択しています。

5-2. 実機動作による性能の計測

ターゲット・ボードでの動作を確認し、Performance Counter を使って演算時間(サイクル数)を計測した結果を 下記に示します。

演算	カスタム・インストラクション(clk)	ソフトウェア演算(clk)
加算	29	218
減算	29	239
乗算	25	236
除算	46	2151
整数⇒倍精度浮動小数型変換	19	83
倍精度浮動小数⇒整数型変換	20	37
比較(<)	19	54
比較(<=)	19	54
比較(>)	19	51
比較(>=)	19	51
比較(==)	19	38
比較(!=)	19	38

※ 比較のため、演算実行時のデータ設定処理等を含んでいます。

※ オプティマイズ・レベルは "-00" (オプティマイズ無し) で確認しています。

5-3. 使用リソース

各演算コアとそれらの合計の使用リソースを下記に示します。

コア	ALMs	ALMs used memory	Block memory bits	DSP Block
ALT_FP_FUNCTION(ADD)	661.7	10	0	0
ALT_FP_FUNCTION(MUL)	148.3	0	0	4
ALT_FP_FUNCTION(DIV)	796.4	110	389120	15
ALT_FP_FUNCTION(I2D)	155.4	0	0	0
ALT_FP_FUNCTION(D2I)	93.9	0	0	0
ALTFP_COMPARE	56.2	0	0	0



6. <u>補足</u>

6-1. 注意事項

- IP コアとソフトウェア演算とは演算アルゴリズムが異なります。したがって、両者を比較した場合、演算結果に若干の誤差が発生したり、非正規化数や無限大、NaN を含む演算で異なった演算結果を出力することがありますのでご注意ください。
- デバイスや IP コアの設定によっては、本資料で紹介している遅延時間を設定できない場合がありますので、その場合は適宜トップ・レベル・モジュールを修正してください。



改版履歴

Revision	年月	概要
1	2018 年 8 月	初版

免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
 株式会社マクニカ アルティマ カンパニー <u>https://www.alt.macnica.co.jp/</u> 技術情報サイト アルティマ技術データベース <u>http://www.altima.jp/members/</u>
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。