プロセッサ DLX integer 仕様

PEAS Project

平成 14 年 10 月 30 日

本ドキュメントは，チュートリアル用に作成したプロセッサ DLX integer 仕様書です．ASIP Meister で設計することを前堤としています.

1 設計目標

面積	遅延	消費電力
30000 ゲート	13 ns	45000 μ W/MHz

面積を優先して設計する.

2 パイプライン構成

- ５段パイプライン
  1. 命令フェッチ
  2. 命令デコード / レジスタ読み出し
  3. 演算実行
  4. メモリアクセス
  5. レジスタ書き込み
- 遅延分岐あり

  遅延分岐スロット数 1
- 命令幅

  32 ビット
- データ幅

  32 ビット

3　使用リソース

ASIPmeister では、“プログラムカウンタ”，“命令レジスタ”，“命令メモリアクセスユニット”，“データメモリアクセスユニット”，“レジスタファイル” の使用を想定しているためそれぞれのリソースを用意する．

また，演算用に算術論理演算ユニット (以下 ALU)，符号拡張用に符号拡張ユニット (以下 EXT) を用意する．乗算，除算，シフト等の演算を行なうための，乗算ユニット (以下 MUL)，除算ユニット (DIV)，シフタ (以下 SFT) を追加する．無条件分岐で使用するため，符号拡張ユニットを 1 つ余計に用意しておく．

各リソースは 32 ビットデータを処理可能なものを選択する．

用途	選択する FHM	リソースの名称	パラメタ
プログラムカウンタ	pcu	PC	増加ステップ幅=4 加算アルゴリズム=cla(桁上げ先見加算)
命令レジスタ	register	IR	
命令メモリ アクセスユニット	imau	IMAU	データのビット幅 = 32 ビット アドレス空間 = 32 ビット
データメモリ アクセスユニット	dmau	DMAU	データのビット幅 = 32 ビット アドレス空間 = 32 ビット データのアクセス幅 = 8 ビット
レジスタファイル	registerfile	GPR	レジスタ本数=32 読み出しポート数=2 書き込みポート数=1
ALU	alu	ALU0	加算アルゴリズム=cla
EXT	extender	EXT0	入力データ幅=16 ビット
MUL	multiplier	MUL0	乗算アルゴリズム = seq 加算アルゴリズム = cla データタイプ = 2 の補数
DIV	divider	DIV0	除算アルゴリズム = seq 加算アルゴリズム = cla データタイプ = 2 の補数
SFT	shifter	SFT0	シフト量 = 可変
EXT	extender	EXT1	入力データ幅 = 28 ビット

4　入出力ポート

設計するプロセッサコアに供給される” クロック”，“リセット信号” を宣言する必要がある．

(FHM-DB から選択した) 使用するデータメモリ・アクセスユニットでは，“アドレスとデータをやりとりするバス” と，メモリとのハンドシェイクに使用する”acknowledge 信号”，“request 信号” をやりとりするポートが必要となる．また，書き込みモードを指定するポートも必要となる．命令メモリ・アクセスユニットでは,” アドレスとデータをやりとりするバス” のみ必要となる．

ポート名	信号の向き	データ幅 (ビット)	用途
CLK	in	1	クロック信号
Reset	in	1	リセット信号
instAB	out	32	命令メモリ用アドレスバス
instDB	in	32	命令メモリ用データバス
DataAB	out	32	データメモリ用アドレスバス
DataDB	inout	32	データメモリ用データバス
DataAck	in	1	データメモリ用 acknowledge 信号
DataReq	out	1	データメモリ用 request 信号
dataWm	out	4	データメモリ用書き込みモードバス

5　命令セット

本節では dlx_integer の 4 種類の命令形式を示した後，個々の命令の概要と命令コード一覧を示す.

5.1　命令形式

命令形式は以下の４種類を用意する.

1. レジスタ-レジスタ形式 (RR 形式)

   レジスタから 2 つのデータを読みだして処理を行ない，結果を他のレジスタに書き込む命令で使用.

2. レジスタ-即値形式 (RI 形式)

   レジスタから 1 つ，即値を 1 つのデータとして処理を行ない，結果を他のレジスタに書き込む命令で使用．メモリ・アクセスの命令でも使用.

3. 条件分岐形式 (B 形式)

   レジスタから 1 つのデータを読みだしてゼロとの比較を行ない，条件が真であれば分岐する命令で使用．分岐先アドレスは 16 ビットオフセット PC 相対となる.

   (実際は，レジスタ-即値形式と同じ形式．)

4. 無条件分岐形式 (J 形式)

   無条件分岐の命令で使用．分岐先アドレスは 26 ビットオフセット PC 相対となる.

5.2　命令概要

命令名	命令の意味		命令形式
ADD	add	符号つき加算	RR 形式
ADDU	add Unsigned	符号なし加算	RR 形式
ADDI	add immediate	符号つき即値加算	RI 形式
ADDUI	add unsigned immediate	符号なし即値加算	RI 形式
SUB	subtarct	符号つき減算	RR 形式
SUBU	subtract unsigned	符号なし減算	RR 形式
SUBI	subtract immediate	符号つき即値減算	RI 形式
SUBUI	subtract unsigned immediate	符号なし即値減算	RI 形式
MULT	multiply	符号つき乗算	RR 形式
MULTU	multiply unsigned	符号なし乗算	RR 形式
DIV	divide	符号つき除算	RR 形式
DIVU	divide unsigned	符号なし除算	RR 形式

表 1: 命令概要 1

命令名	命令の意味		命令形式
AND	and	論理積	RR 形式
ANDI	and immediate	論理積	RI 形式
OR	or	論理和	RR 形式
ORI	or immediate	論理和	RI 形式
XOR	exclusive or	排他的論理和	RR 形式
XORI	exclusive or immediate	排他的論理和	RI 形式
SLL	shift left logical	左論理シフト	RR 形式
SRL	shift right logical	右論理シフト	RR 形式
SRA	shift right arithmetic	右算術シフト	RR 形式
SLLI	shift left logical immediate	左論理シフト	RI 形式
SRLI	shift right logical immediate	右論理シフト	RI 形式
SRAI	shift right arithmetic immediate	右算術シフト	RI 形式
SLT	set less than	未満ならセット	RR 形式
SGT	set greater than	より大きければセット	RR 形式
SLE	set less equal	以下ならセット	RR 形式
SGE	set greater equal	以上ならセット	RR 形式
SEQ	set equal	同じならセット	RR 形式
SNE	set not equal	異なればセット	RR 形式
SLTI	set less than immediate	即値未満ならセット	RI 形式
SGTI	set greater than immediate	即値より大きければセット	RI 形式
SLEI	set less equal immediate	即値以下ならセット	RI 形式
SGEI	set greater equal immediate	即値以上ならセット	RI 形式
SEQI	set equal immediate	即値と同じならセット	RI 形式
SNEI	set not equal immediate	即値と異なればセット	RI 形式

表 2: 命令概要 2

命令名	命令の意味		命令形式
LHI	load high immediate	レジスタの上位に即値をロ-ド	RI 形式
LB	load byte	符号拡張したバイトデ-タのロ-ド	RI 形式
LBU	load byte unsigned	ゼロ拡張したバイトデ-タのロ-ド	RI 形式
LH	load halfword	符号拡張したハ-フワ-ドデ-タのロ-ド	RI 形式
LHU	load halfword unsigned	ゼロ拡張したハ-フワ-ドデ-タのロ-ド	RI 形式
LW	load word	ワ-ドデ-タのロ-ド	RI 形式
SB	store byte	バイトデ-タのストア	RI 形式
SH	store halfword	ハ-フワ-ドデ-タのストア	RI 形式
SW	store word	ワ-ドデ-タのストア	RI 形式
BEQZ	branch equal zero	ゼロと等しければ分岐	B 形式
BNEZ	branch not equal zero	ゼロと異なれば分岐	B 形式
J	jump	無条件 PC 相対分岐	J 形式
JAL	jump and link register	戻りアドレスを待避する無条件 PC 相対分岐	J 形式
JR	jump register	無条件レジスタ間接分岐	RR 形式
JALR	jump and link register	戻りアドレスを待避する無条件レジスタ間接分岐	RR 形式

表 3: 命令概要 3

5.3 命令コード一覧

命令名	オペコ-ド 31..26	レジスタ 1 25..21	レジスタ 2 20..16	レジスタ 3 15..11	オペコ-ド 10..0
SLL	000000	rs0	rs1	rd	00000000000
SRL	000000	rs0	rs1	rd	00000000010
SRA	000000	rs0	rs1	rd	00000000011
JR	000000	rs0	rs1	rd	00000001000
JALR	000000	rs0	rs1	rd	00000001001
MULT	000000	rs0	rs1	rd	00000011000
MULTU	000000	rs0	rs1	rd	00000011001
DIV	000000	rs0	rs1	rd	00000011010
DIVU	000000	rs0	rs1	rd	00000011011
ADD	000000	rs0	rs1	rd	00000100000
ADDU	000000	rs0	rs1	rd	00000100001
SUB	000000	rs0	rs1	rd	00000100010
SUBU	000000	rs0	rs1	rd	00000100011
AND	000000	rs0	rs1	rd	00000100100
OR	000000	rs0	rs1	rd	00000100101
XOR	000000	rs0	rs1	rd	00000100110
SLT	000000	rs0	rs1	rd	00000101010
SGT	000000	rs0	rs1	rd	00000101011
SLE	000000	rs0	rs1	rd	00000101100
SGE	000000	rs0	rs1	rd	00000101101
SEQ	000000	rs0	rs1	rd	00000101110
SNE	000000	rs0	rs1	rd	00000101111

表 4: RR 形式

命令名	オペコード 31..26	即値 25..0
J	000010	const
JAL	000011	const

表 5: J 形式

命令名	オペコ-ド 31..26	レジスタ 1 25..21	レジスタ 2 20..16	即値 15..0
BEQZ	000100	rs0	rs1	const
BNEZ	000101	rs0	rs1	const
ADDI	001000	rs0	rs1	const
ADDUI	001001	rs0	rs1	const
SUBI	001010	rs0	rs1	const
SUBUI	001011	rs0	rs1	const
ANDI	001100	rs0	rs1	const
ORI	001101	rs0	rs1	const
XORI	001110	rs0	rs1	const
LHI	001111	rs0	rs1	const
SLLI	010000	rs0	rs1	const
SRLI	010001	rs0	rs1	const
SRAI	010010	rs0	rs1	const
SLTI	011010	rs0	rs1	const
SGTI	011011	rs0	rs1	const
SLEI	011100	rs0	rs1	const
SGEI	011101	rs0	rs1	const
SEQI	011110	rs0	rs1	const
SNEI	011111	rs0	rs1	const
LB	100000	rs0	rs1	const
LH	100001	rs0	rs1	const
LW	100011	rs0	rs1	const
LBU	100100	rs0	rs1	const
LHU	100101	rs0	rs1	const
SB	101000	rs0	rs1	const
SH	101001	rs0	rs1	const
SW	101011	rs0	rs1	const

表 6: RI/B 形式

6　リセット割り込み

プロセッサ dlx_integer では，リセット割り込みを備える．実装するリセット割り込みは以下のようなものとする．

- 発生条件

  reset 属性のポート Reset から信号”1” が入力されることで発生する．

  (現在の ASIP Meister では、ここで示した発生条件のみに対応．)

- リセット処理

  リソース PC(プログラムカウンタ), IR(命令レジスタ), GPR(汎用レジスタファイル) をそれぞれリセットする．

- 処理サイクル数

  １サイクル