

情報科学実験C 中間レポート

氏名 山久保孝亮
所属 大阪大学基礎工学部情報科学科ソフトウェア科学コース
メールアドレス u327468b@ecs.osaka-u.ac.jp
学籍番号 09B22084
提出日 2024 年 12 月 10 日

1 Tiny-Processor と C-Processor のデータパスの違い

以下の図 1,2 はそれぞれ Tiny-Processor と C-Processor のデータパスである。

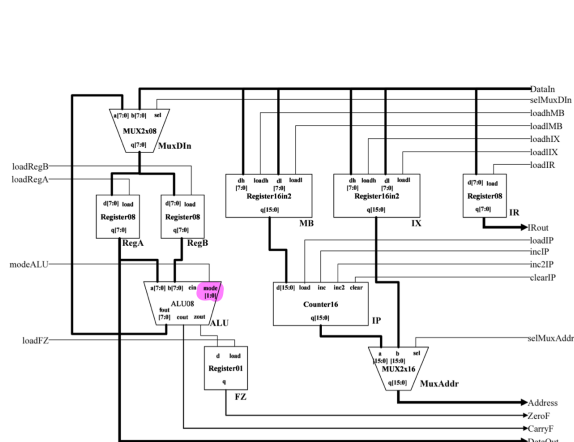


図 1: Tiny-Processor のデータパス

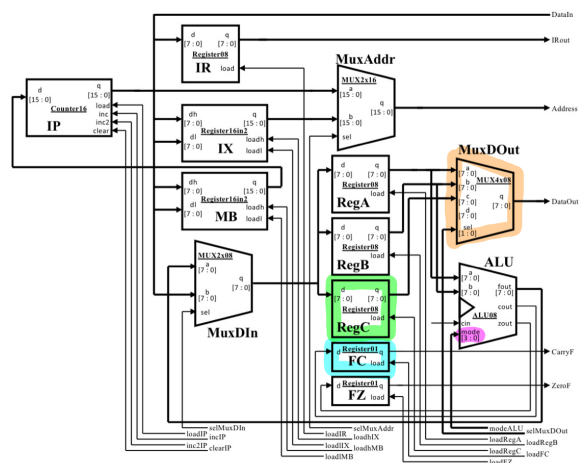


図 2: C-Processor のデータパス

この二つのデータパスの違いを明確にするために色付けをした。以下でそれぞれの違いについて述べる。

- 図 1 のピンクに塗った部分は ALU の入力 mode を表す。Tiny-Processor では mode は 2 ビットの入力として設計されている。図 2 のピンクで塗った部分は C-Processor の同じく ALU の入力 mode である。C-Processor では Tiny-Processor よりも機能が増えており、mode の入力が増加している。
- 図 2 の緑で塗った部分は RegC であり、C-Processor で追加されている。入力は新たに追加された loadRegC 以外は RegA, RegB と同じである。このレジスタ C は STDI 命令の際に使用される。詳細な説明は 2 で記述する。
- 図 2 の青色で塗った部分は FC であり、C-Processor で追加されている。入力は新たに追加された loadFC と ALU の出力である cout で、出力は q 即ち CarryF である。これは ALU の演算の結果、桁上りが発生したかを判定する。これから CarryF という信号が得られ、JPC 命令に利用される。
- 図 2 のオレンジで塗った部分は MuxDOOut であり、4 入力のマルチプレクサである。ただし、4 入力の内 3 つしか使っておらずその 3 つの入力は RegA, RegB, RegC の出力である。また、4 つの入力の内どの値を出力するかを決定する 2 ビットの selMuxDOOut を追加した。そして出力が DataOut となる。Tiny-Processor では RegA の出力が DataOut となっていたが、C-Processor では三つのレジスタからマルチプレクサで選択して DataOut を決定している。

2 STDI 命令のデータの流れ

以下の図 3 は 1 の図 2 で示した C-Processor のデータパスに、STDI 命令を実行する際のデータの流れを追加したものである。

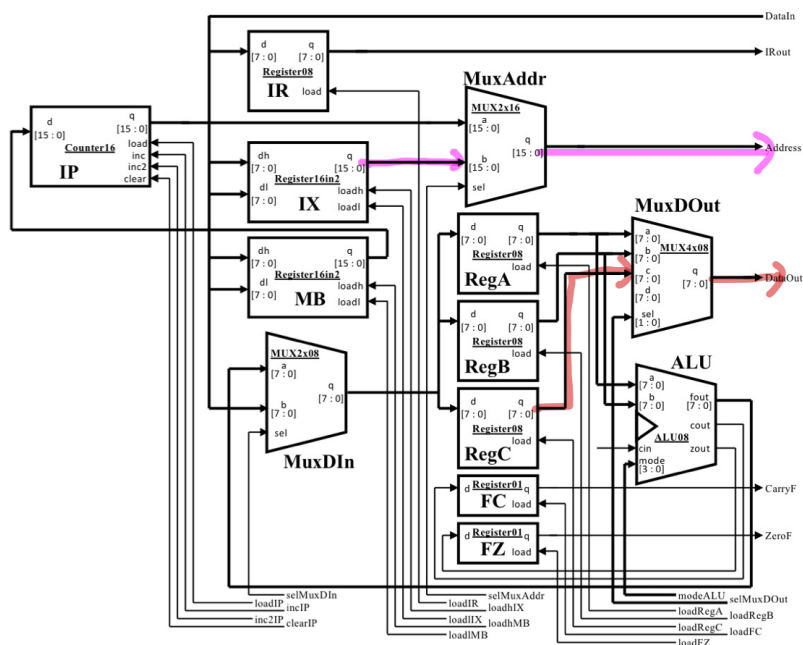


図 3: C-Processor における STDI 命令のデータの流れ

ピンクの矢印はアドレス、赤の矢印はデータの流れを表している。

3 外部出力信号用ジョンソンカウンタと内部制御信号用ジョンソンカウンタ

設計に用いた外部出力信号のジョンソンカウンタと内部制御信号のジョンソンカウンタを示し、それぞれ説明する。

3.1 外部出力用ジョンソンカウンタ

C-Processor の外部出力用ジョンソンカウンタは以下になる。

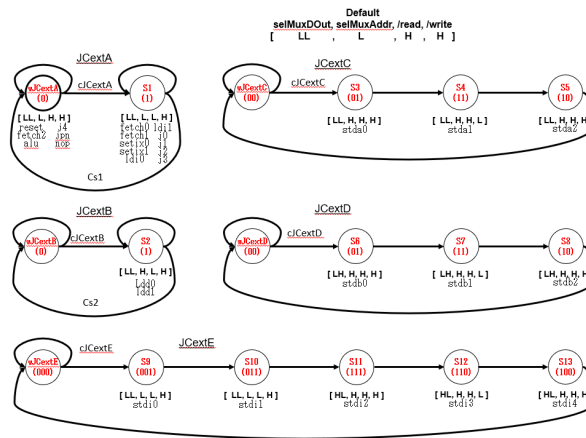


図 4: 外部出力用ジョンソンカウンタ

3.2 内部出力用ジョンソンカウンタ

C-Processor の内部出力用ジョンソンカウンタは以下になる。

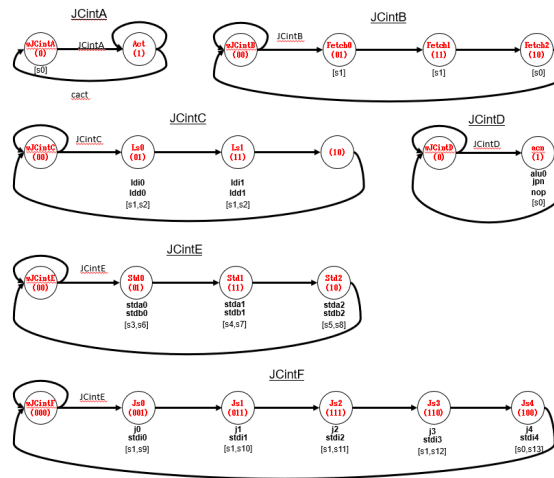


図 5: 内部出力用ジョンソンカウンタ

4 工夫点

C-Processor の設計にあたって、何かしらの工夫を行なったことがあれば、記述する。拡張課題は以下の様に、それぞれの節を用意して説明する。

5 拡張課題

5.1 拡張課題 a：プロセッサ命令セットの拡張

今回私が追加した命令セットは論理シフトである．いかにその仕様を示す．

命令	オペランド	サイズ	コード
SLLA	id		
SRLA	id		
SLLB	id		
SRLB	id		

表 1: 追加した命令

この 4 命令は，第一オペランドで即値により指定した数値分レジスタ A またはレジスタ B を論理シフトするものである．これは ALU の演算の処理を切り替える 4 ビットのベクトル modeALU で未使用の部分を使用して実装した．したがって，このシフト命令を追加した後の modeALU の処理は以下のようになる．

LLLL	A + B	LHLL	not A	HLLL	not B	HHLL	
LLLH	A - B	LHLH	A + 1	HLLH	B + 1	HHLH	
LLHL	A and B	LHHL	A - 1	HLHL	B - 1	HHHL	
LLHH	A or B	LHHH	id	HLHH	id	HHHH	

表 2: シフト命令を追加した後の modeALU の処理

5.2 拡張課題 b：簡易アセンブラの作成

5.3 拡張課題 c：演算機の改善