

具迴圈加速機制之處理器

參展人員：趙姝戎、何宗憲

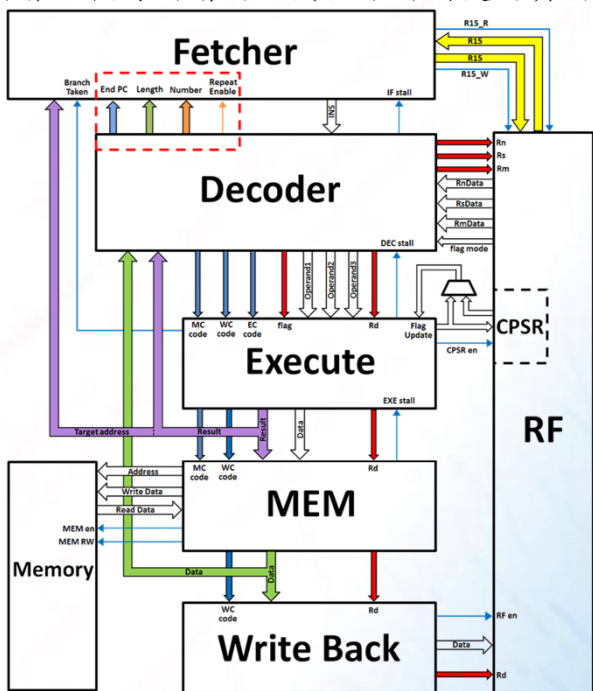
指導教授：邱日清

一. 動機

由於在執行迴圈程式的跳躍指令需要判斷執行幾次以及何時跳出迴圈，因此在判斷迴圈是否繼續執行或者跳出迴圈時會造成時間上的浪費。故本專題以處理器使用效率的提升為目的，新增迴圈指令，使PC能重複地在一個範圍內循環直至迴圈結束以減少時間的浪費。

二. 架構介紹

處理器架構參考ARM7相容指令集，設計簡易五階管線處理器，額外增加硬體(Repeat Unit)於指令抓取階層以執行新增的迴圈指令解碼後的資料。



圖一 設計之簡易五階管線處理器架構

在設計處理器前須從ARM7相容指令集未被使用的指令集中訂出迴圈指令之格式：

[illegible]

1	Number Imm [7:0]				Length Imm[3:0]		End PC Imm[7:4]		1	00		1	End PC Imm[3:0]	
24	23	16	15	12	11	8	7	6	5	4	3	2	1	0

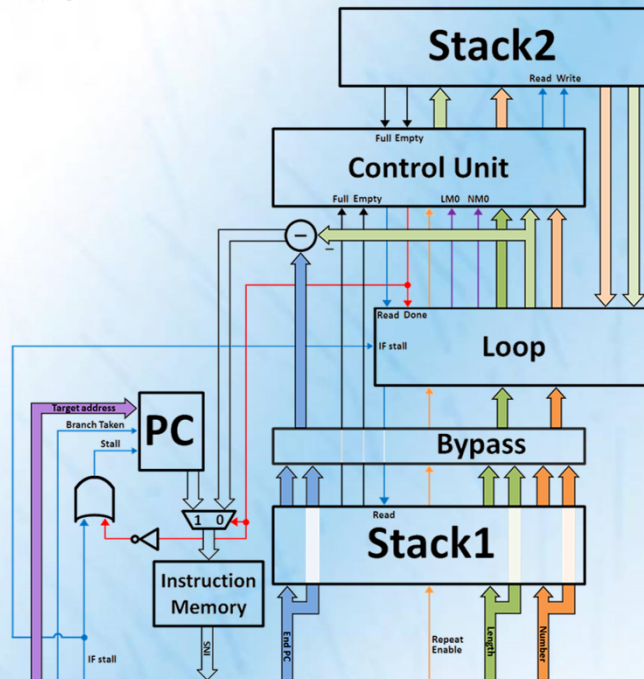
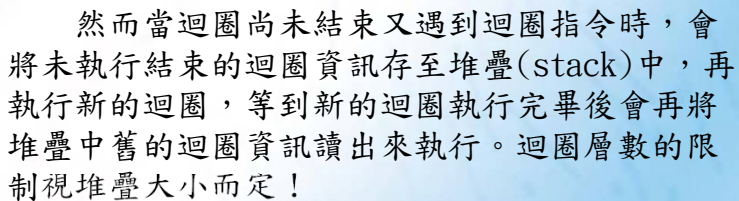
0	0000			RNUM	RLEN		0000		1	00		1	RDC	
24	23	20	19	16	15	12	11	8	7	6	5	4	3	0

圖二 迴圈指令格式

迴圈指令會在指令解碼(Instruction Decoder)階層解碼出以下資訊：

1. 迴圈的終止PC(End PC)
2. 迴圈的長度(Length)
3. 迴圈的重複次數(Number)
4. 解碼出迴圈指令(Repeat Enable)

Repeat Unit根據迴圈指令解碼後的結果使PC能重複地在一個範圍內循環直至迴圈結束，再透過返回PC跳出迴圈。



圖二 Instruction Fetch架構

接下來介紹指令抓取階層的Repeat Unit，其架構分成4部分介紹：

1. 迴圈指令解碼資訊堆疊區(Stack1以及Bypass)
2. PC迴圈運算區(Loop)
3. 控制區(Control)
4. 迴圈指令暫存堆疊區(Stack2)

1. 迴圈指令解碼資訊堆疊區：

指令解碼階層解碼出的資料會存至堆疊，但為了能及時資料送往PC迴圈運算區而在存至堆疊前將資料bypass至PC迴圈運算區。

2. PC迴圈運算區：

根據控制線決定迴圈運算的資料來源。資料來源有三種：第一是迴圈指令解碼資訊堆疊區的起始資訊；第二是PC迴圈運算的運算結果；第三是迴圈指令暫存堆疊區。

3. 控制區：

根據堆疊的Empty或是Full、Repeat Enable以及迴圈長度和次數是否等於零(LM0、NM0)運算出現在迴圈的狀態及是否需要將堆疊的資料取出或寫入迴圈指令暫存堆疊區的資料。

4. 迴圈指令暫存堆疊區：

執行迴圈時若有新的迴圈指令出現就會把正在執行的迴圈資訊存至堆疊，直至新的迴圈結束後才會將舊的堆疊資料取出，迴圈最多能有八層。