

Lab 10

LZ77 Decoder

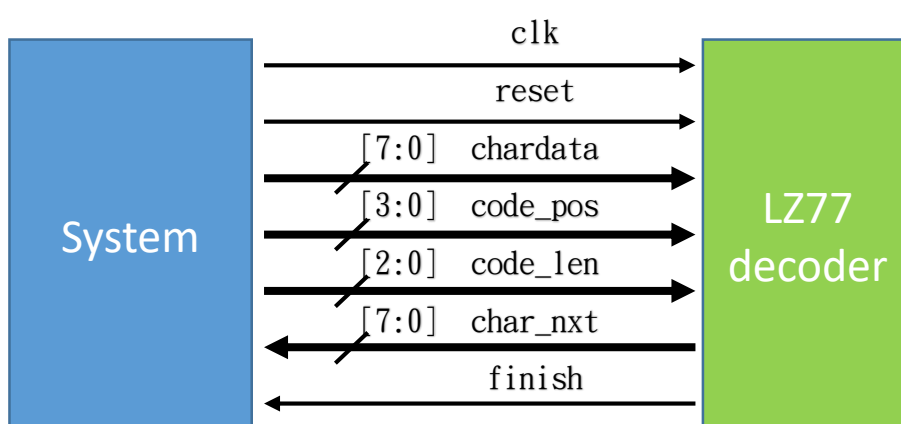
1. 問題描述

Ziv和Lempel於1977年發表題為「順序數據壓縮的一個通用算法（A Universal Algorithm for Sequential Data Compression）」的論文，論文中描述的算法被後人稱為LZ77算法。

LZ77是典型的基于字典的無損壓縮算法，現在很多壓縮技術都是基于LZ77，它將長字符串（也稱為短語）編碼成短小的標記，用小標記代替字典中的短語，從而達到壓縮的目的。也就是說，它通過用小的標記來代替數據中多次重複出現的長串方法來壓縮數據。其處理的符號不一定是文本字符，可以是任意大小的符號。

2. 設計規格

2.1 系統方塊圖



圖一、系統方塊圖

2.2 輸入/輸出介面

表一、輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	本系統為同步於時脈正緣之同步設計。
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
code_pos	I	4	匹配字符串的開始字符串與待編碼緩沖區的距離。
code_len	I	3	匹配字符串的長度。
chardata	I	8	待編碼區下一個等待編碼的字符。
char_nxt	O	8	完成的編碼。
finish	O	1	有效的輸出訊號，為high時，表示目前的Decoder動作已經全部完成。

2.3 LZ77 演算法描述

LZ77是典型的基于字典的無損壓縮算法，主要有兩個特點：

1. 它利用編碼的輸入序列作為字典。
2. 使用滑動視窗對數據進行編碼。滑動視窗由 search buffer 和 look-ahead buffer 組成。Search buffer 保存已編碼的輸入序列，look-ahead buffer 存儲要編碼的序列。

以下是 LZ77編碼的演算法步驟：

1. 將指針移動到 search buffer 中的數據，直到等於 look-ahead buffer 中的第一個字符。然後檢查 search buffer 和 look-ahead buffer 中的下一個字符。如果它們相等，則再次檢查下一個字符，直到它們不相等。相等字符的總數稱為匹配長度。
2. 重複步驟 1，直到確定所有匹配序列中最長的一個。
3. 輸出編碼結果[offset, match_len, char_nxt]。
 1. offset：匹配字符串中第一個字符的位置到search buffer首位的偏移量。
 2. match_len：最長匹配序列的長度。
 3. char_nxt： look-ahead buffer 中最長匹配序列後面的字符。
4. 將 match_len + 1 個字符從look-ahead buffer移動到search buffer 。
5. 重複步驟 1 到步驟 4，直到所有數據都被編碼。

※LZ77解碼演算法與編碼算法相反。

LZ77編碼演算法範例：假設序列為「112a112a2112112112a21\$」，search buffer 的大小為 9 個字符長，look-ahead buffer 為 8 個字符長。而「\$」為輸入序列的終止字元。

	search buffer	look-ahead buffer	input sequence	matched string	encode result
Round1		112a112a	2112112112a21\$		[0, 0, 1]
Round2	1	12a112a2	112112112a21\$	1	[0, 1, 2]
Round3	112	a112a211	2112112a21\$		[0, 0, a]
Round4	112a	112a2112	112112a21\$	112a	[3, 4, 2]
Round5	112a112a2	11211211	2a21\$	112	[8, 3, 1]
Round6	112a2112112a21	\$		12112	[2, 5, a]
Round7	12112112a	21\$		21	[7, 2, \$]

圖二、LZ77編碼器步驟

編碼結果為：

{[0, 0, 1]、[0, 1, 2]、[0, 0, a]、[3, 4, 2]、[8, 3, 1]、[2, 5, a]、[7, 2, \$]}

注意，在Round 6中，由於LZ77算法的滑動視窗是由 search buffer 和 look-ahead buffer 組成的，所以要編碼的字符不僅可以在 search buffer 中搜索，還可以在 look-ahead buffer 中搜索。

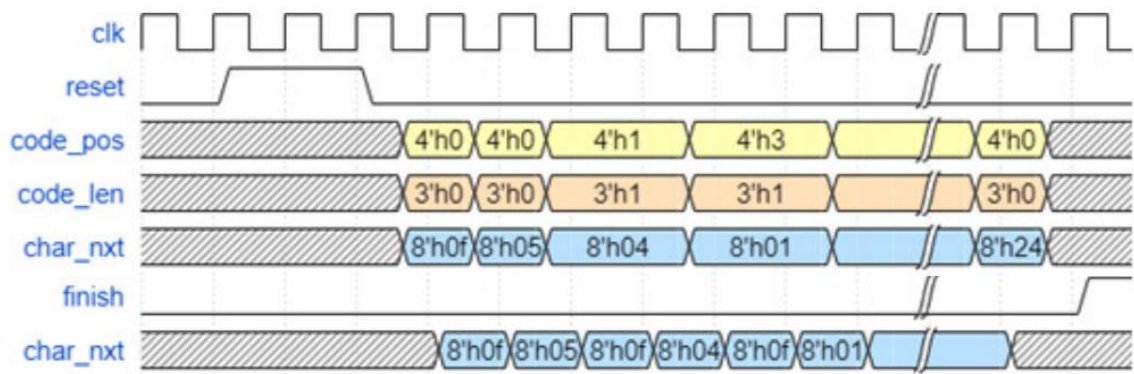
LZ77解碼演算法和編碼過程相反：

	decode input	decode result	search buffer	look-ahead buffer
Round1	[0, 0, 1]	1	<u>1</u>	
Round2	[0, 1, 2]	12	112	
Round3	[0, 0, a]	a	<u>112a</u>	
Round4	[3, 4, 2]	112a2	<u>112a</u> 112a2	
Round5	[8, 3, 1]	1121	112a21 <u>121</u>	<u>??</u>
Round6	[2, 5, a]	12112a	<u>121</u> 12112a	
Round7	[7, 2, \$]	21	112112a21	

圖三、LZ77解碼器步驟

2.3 資料輸入與結果輸出描述

- search buffer 的大小為 9 個字符長，
- look-ahead buffer 的大小為 8 個字符長。
- 在解碼完成後的 5 個 cycle 內，應該送出 finish 訊號為 high。
- tb會在reset完成後立即發送解碼數據(code_pos, code_len, chardata)，decoder應該輸出解碼結果(char_nxt)，輸出時序如下。



圖四、資料輸入與結果輸出

3. 評分標準

本次作業評分方式會依照設計完成程度和完成時間，分為 A、B、C、D 四種等級。作業最多兩人一同協作完成，若三人以上程式碼雷同，則三人皆視為等級 D，兩人只須要有一人繳交程式碼即可，請在程式碼前面註明兩人的姓名學號。結報(每人)內容須包含協作者名稱、程式貢獻比例、程式面積、設計心得、遭遇的困難和解決的方法。

◆ 等級 A：100 分

等級 A 條件：

- a、在時限內完成。
- b、通過防抄襲比對。

◆ 等級 B：80 分

等級 B 條件：

- a、在時限內完成。
- b、使用範例程式碼(截止日前三天公布)完成。

◆ 等級 C：60 分

等級 C 條件：

- a、在截止後一周內完成。
- b、使用範例程式碼(截止日前三天公布)或自己撰寫，並通過防抄襲比對。

◆ 等級 D：0 分

等級 D 條件：符合下列任一項

- a、未繳交程式。
- b、繳交程式碼功能不齊全。
- c、未通過防抄襲比對。

3.1 程式面積

將完成的程式，使用 Quartus 軟體合成，並記錄下程式使用的邏輯閘和暫存器數量。

Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	764 / 49,760 (2 %)
Total registers	289
Total pins	36 / 360 (10 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	2 / 288 (< 1 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

圖五、Quartus合成的數據報告

4. 附錄

附錄4.1~3為設計檔案說明；

4.1 設計檔案說明

表二、設計檔案說明

檔名	檔案說明
lab10.v	所有學生使用的設計檔，包含輸入輸出宣告。
lab10tb.v	TestBench檔案。
data*.dat	測資資料。

4.2 測試檔內容

本次練習所提供的testbench檔案和設計檔，有多增加幾行特別用途的敘述：

```
“lab10tb.v”
`define PAT "data1.dat"
`define End_CYCLE 1000000

-----
“lab10.v”
module lab10(.....);
...
for(i = 0; i < `length; i = i+1)
    $dumpvars(1, count[i]);
...

執行測試檔時，請執行以下指令

cmd> iverilog -g2012 -o temp lab10tb.v
```

4.3 特殊情況

若在執行的過程中出現以下警告，可以忽略：

```
VCD info: dumpfile Lab.fsdb opened for output.
VCD warning: array word Pin_Name will conflict with
an escaped identifier.
```