

2006【ARM Code-O-Rama】設計大賽

題目說明

<JPEG-LS Decoder最佳化設計>

一、問題描述

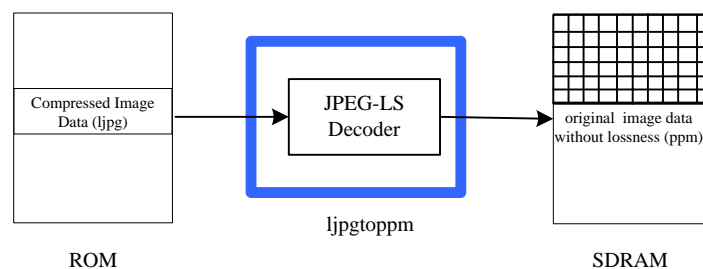
JPEG-LS (Lossless) 是一種非失真與近乎非失真的影像壓縮標準，主要應用在醫學影像的傳送及儲存上，有關詳細的JPEG-LS壓縮技術請參考 (Reference [1][2][3][4][5])。本次競賽將會提供各參賽隊伍JPEG-LS壓縮(Encoder)、解壓縮(Decoder)的原始碼(Source Code)及Makefile，此原始碼可由GNU Compiler Collection (GCC) 編譯連結成可執行檔。

本次競賽主要是針對JPEG-LS解壓縮部分。參賽者應將解壓縮之原始碼移植至ARM的軟體發展環境ARM Development Suite (ADS) 上，並透過AXD Debugger中Load Memory From File功能，將欲解壓縮之影像 (input data)，放置於記憶體某特定區段中，並修改解壓縮程式，使其至某特定記憶體位置，讀取影像資料，進行影像解壓縮處理，並將處理後之影像 (output data) 存到記憶體某特定位置上。最後各位參賽者，可利用AXD Debugger中Save Memory to File功能，將存放在記憶體區段之解壓縮影像，儲存成檔案，並觀察其結果是否符合預期 (與壓縮前影像比對)。由於嵌入式系統記憶體資源有限 (記憶體大小、速度)，因此程式對記憶體配置及使用方式 (例如：將常用到的變數放置在較快速之記憶體)，將會影響整體系統效能。請各位同學依此特性，配合主辦單位所提供記憶體配置圖 (圖二)，針對此原始碼進程式最佳化調整，來達到最佳的系統效能 (cycle count) 及最少的記憶體成本 (code size)。

二、設計規格

2.1 系統方塊圖

參賽者將JPEG-LS之解壓縮程式ljpgtoppm移植至ADS上，並透過AXD Debugger中Load Memory From File功能，將欲解壓縮之影像 (input data)，放置於ROM的特定位址，並從該位址讀取影像資料，進行影像解壓縮處理，並將處理後之影像 (output data) 存到SDRAM的特定位置上。



圖一 系統方塊圖

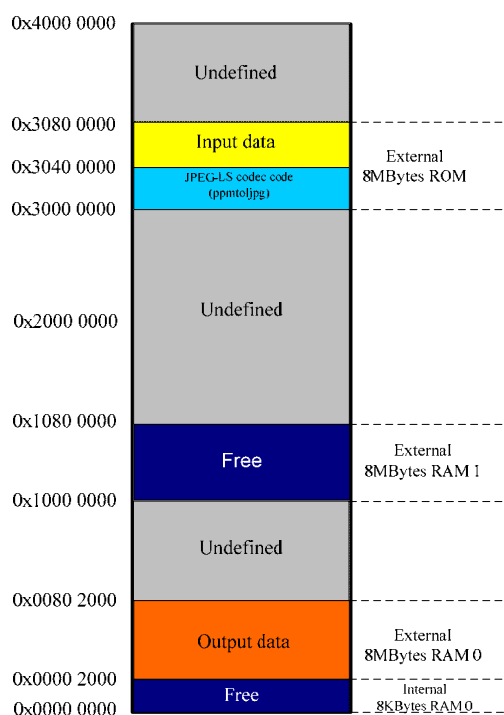
2.2 系統規格說明

本次競賽。參賽者應修改 JPEG-LS 套件，使其可以透過 ADS 編譯。並可將編譯結果載入且運行於 ARM 指令集層級的模擬器 (ARMulator) 中。ARMulator 主要提供參賽者驗證編譯出來的映像檔 (image)，其限定的處理器設定組態可對照如下表一：

表一 Target processor configuration

Processor Variant	ARM9TDMI-REV2
Clock	200Mhz
Memory Map File	使用主辦單位提供之 mapfile (m1.map)
其他設定	使用預設值

此外，藉由引入記憶體配置描述檔 (mapfile) (參照本試題卷之頁3)，本競賽亦限定記憶體的屬性及使用方式，各種記憶體的起始位址、長度以及使用方式如圖二所示：



圖二 記憶體配置圖

接著，參賽者將使用主辦單位提供的 JPEG-LS 套件來完成本競賽，安裝方式可參照本試題卷之頁5。此 JPEG-LS 套件主要分為兩個模組：ppmtoljpg 及 ljpgtoppm，本競賽僅使用 ljpgtoppm 模組來做解壓縮功能。為了使其正常運行於如圖二的記憶體配置，請務必做必要的修改，使之符合如下表二之規定：

表二 記憶體資源使用說明

記憶體種類	起始位址	大小	資源使用規定
ROM	0x30000000	4MBytes	程式的映像檔
ROM	0x30400000	4MBytes	Input data
SDRAM0	0x00002000	8MBytes	Ouput data
SDRAM1	0x10000000	8Mbytes	自由使用
Internal SRAM	0x00000000	8KBytes	自由使用

2.3 輸入檔案描述

2.3.1 影像檔

PPM [6] (Portable Pixel Map) 為一個簡單的非壓縮性圖檔格式，用來作為ppmtoljpg模組的輸入檔，其內容定義了兩個部分:圖像資訊及像點 (pixel) 資料。其中，圖像資訊的第一行為magic number，用以判別像點資料的儲存方式，主要有P3 (ASCII data)、P6 (binary data) 及 P2 (grey data) 三種格式。另外，第四行 (intensity) 定義了可使用的色階，其範圍可為0~255。本競賽使用P6格式的ppm圖檔如下表三所示，以未經修改的ppmtoljpg 壓縮後，作為驗證參賽者ljpgtoppm模組的輸入檔，且此輸入檔的大小以不超過ROM的大小 (4MBytes) 為原則。

表三 1 x 3 P6 格式 ppm sample

P3	(magic number)
#comment line	(comment line)
1 3	(width and height)
255	(intensity)
FFFFFFFF00000000FF	(定義了三個點的 RGB 值：白色 - FFFFFFFF, 紅色 - FF000000, 藍色 - 0000FF)

LJPG為PPM 經JPEG-LS演算法壓縮後所產生的檔案格式，其詳細的內容請參考相關資料[5]。

2.3.2 記憶體描述檔

本競賽所使用的記憶體配置描述檔如下表四 (mapfile)，共定義了四塊記憶體區域，詳細的設定範例可參考ADS Debug Target Guide [7]。

表四 memory map file

00000000	2000	ISRAM	4	rw	1/1	1/1
00002000	800000	SDRAM0	4	rw	100/60	100/60

```

10000000 800000 SDRAM1 4 rw 100/60 100/60
30000000 800000 ROM 2 r 150/100 150/100
00000000 40000000 DUMMY 4 - 1/1 1/1

```

- ROM：唯讀，起始位址為0x30000000，大小為8MBytes，資料匯流排寬度為16bit，讀取跟寫入的記憶體存取時間分別為 150/100 ns（非連續存取/連續存取）及150/100 ns（因為唯讀，所以關於寫入的數值忽略不使用）。
- External SDRAM 0：可讀寫，起始位址為0x00002000，大小為8MBytes，資料匯流排寬度為32bit，讀取跟寫入的記憶體存取時間分別為150/100 ns（非連續存取/連續存取）及100/60 ns。
- External SDRAM 1：可讀寫，起始位址為0x10000000，大小為8MBytes，資料匯流排寬度為32bit，讀取跟寫入的記憶體存取時間分別為 150/100 ns（非連續存取/連續存取）及100/60 ns。
- Internal SRAM：可讀寫，起始位址為0x0，大小為8KBytes，資料匯流排寬度為32bit，讀取跟寫入的記憶體存取時間分別為 1/1 ns（非連續存取/連續存取）及1/1 ns。
- 存取其他區域將導致Data/Prefetch abort 的處理器例外產生。

另外，處理器針對於記憶體存取所產生的等待時間（wait state），取決於記憶體存取時間。例如時脈100Mhz（週期為10 ns）的處理器，當讀取存取時間為60ns的記憶體，則處理器共需花費6個cycle（1 cycle + 5 wait cycle）來完成。

三、設計平台環境

3.1發展環境

表五 發展環境

	Platform	Tool	Language
程式移植前	Solaris、Linux	GNU Compiler Collection	C/C++
程式移植後	Windows 2000、 XP Professional	ARM Development Suite v1.2	C/C++、ARM Assembly

3.2 JPEG-LS套件安裝說明

在下載JPEG-LS的壓縮檔（tar）後，請將該套件上傳至UNIX工作站（已測試過執行環境為Solaris 2.4及Linux 2.4）。解壓縮後，會得到如下表六的檔案列表以及編譯用之Makefile。在shell下達make指令後，將會得到pnmtoljpg及ljpgtopnm兩個檔案；這兩個檔案分別是JPEG-LS 壓縮及解壓縮的可執行檔。我們可以利用參數引入的方式，將測試影像輸入至此程式內執行，並將結果輸出，使用方式如下所示：

- (a) 將F-18.ppm 這個PPM格式的影像檔進行壓縮，並將壓縮結果存成

```
test.ljpeg
```

```
% ./pnmtoljpg F-18.ppm test.ljpg
```

(b) 將F-18.ljpg 這個JPEG-LS壓縮影像進行解壓縮，並將解壓縮結果存成

```
test.ppm
```

```
% ./ljpgtopnm F-18.ljpg test.ppm
```

表六 JPEG-LS 套件檔案列表

	JPEG-LS Encoder (僅供參考)	JPEG-LS Decoder
Source Code	util.c mcu.c predictor.c pnmtoljpg.c huffc.c write.c pmread.c jpeg.h mcu.h pnmtoljpg.h io.h predictor.h	util.c mcu.c predictor.c ljpgtopnm.c huffd.c read.c jpeg.h mcu.h predictor.h io.h
Test Image	F-18.ppm	F-18.ljpg
Makefile	Makefile	
Map file	m1.map	

四、評分標準

1. 初賽成果報告 (30%)：各參賽者必須繳交初賽報告書 (附件一)，本競賽單位將會依此報告書來進行評分，評分項目詳見 (附件二)。
2. 競賽單位驗證 (30%)：本競賽單位將會驗證各參賽者所繳交之程式碼，並進行評分，其評分項目詳見 (附件四)。

本競賽單位將會採用數組不同Pattern (影像)，透過AXD Debugger中“Load Memory From File..”之功能，將測試影像 (input data) 放置在記憶體位置0x30400000上；經過JPEG-LS解壓縮程式運算處理後，再利用“Save Memory to File..”功能，將存放在記憶體區

段0x00002000之解壓縮影像，儲存成檔案，並與壓縮前的影像，進行點對點比對，如有錯誤則驗證部份給予零分計算。以下為效能評估之計算公式：

$$\text{Performance Index}_{\text{Decoder}} = \left(W_1 \times \frac{(\text{Cycle count}_{\text{Decoder}}^{\text{Worst case}} - \text{Cycle count}_{\text{Decoder}}^{\text{Optimal}})}{\text{Cycle count}_{\text{Decoder}}^{\text{Worst case}}} \right) + \left(W_2 \times \frac{(\text{Code size}_{\text{Decoder}}^{\text{Worst case}} - \text{Code size}_{\text{Decoder}}^{\text{Optimal}})}{\text{Code size}_{\text{Decoder}}^{\text{Worst case}}} \right) \quad (1)$$

影像大小在 320 Pixel × 240 Pixel 的情況下

$$\text{Cycle count}_{\text{Decoder}}^{\text{Worst case}} = 1,668,197,859 \quad \text{Code size}_{\text{Decoder}}^{\text{Worst case}} = 20.49\text{KB}$$

$$W_1 = 0.5 \quad W_2 = 0.5$$

式(1)其變數定義如下：

$\text{Cycle count}_{\text{Decoder}}^{\text{Worst case}}$ ：程式尚未最佳化，以及最壞記憶體配置情況下，影像解壓縮(Decoder)所需要的 Total cycle count；此數值至 AXD Debugger → System Views → Debugger internals → Statistics → Total 中觀察。

$\text{Code size}_{\text{Decoder}}^{\text{Worst case}}$ ：程式尚未最佳化，以及最壞記憶體配置情況下，影像解壓縮(Decoder)所需要的 Total ROM Size；此數值為 Code、RO Data、RW Data 之總和。

$\text{Cycle count}_{\text{Decoder}}^{\text{Optimal}}$ ：程式最佳化後，影像解壓縮所需要的 Total cycle count。

$\text{Code size}_{\text{Decoder}}^{\text{Worst case}}$ ：程式最佳化後，影像解壓縮所需要的 Total ROM Size。

$\text{Performance Index}_{\text{Decoder}}$ ：程式最佳化後，所改善(Total cycle count、Code size)的百分比。所改善百分比越高，則此部分分數越高。

3. 現場報告 (40%)：通過初審之隊伍，必須進行現場報告，其評分項目詳見 (附件三)。

五、參考資料

- [1]. Lossless JPEG Codec : <http://datacompression.info/JPEGLS.shtml>
- [2]. JPEG-LS home page : <http://www.jpeg.org/jpeg/jpegls.html>
- [3]. M. Weinberger, G. Seroussi, G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", Hewlett-Packard Laboratories Technical Report No. HPL-98-193R1, November 1998, revised October 1999. IEEE Trans. Image Processing, Vol. 9, August 2000, pp.1309-1324. PDF Postscript
- [4]. M. Weinberger, G. Seroussi, G. Sapiro, "LOCO-I: A Low Complexity, Context-Based, Lossless Image

- Compression Algorithm," Proc. IEEE Data Compression Conference, Snowbird, Utah, March-April 1996.
- [5]. FCD 14495, Lossless and near-lossless coding of continuous tone still images.
- [6]. PPM Format : <http://www.physics.emory.edu/~weeks/graphics/mkppm.html>
- [7]. "ARM Developer Suite Version 1.2 Debug Target Guide (ARM DUI 0058D)", ARM Ltd., (1999).

2006【ARM Code-O-Rama】設計大賽

<JPEG-LS Decoder最佳化設計>

初 賽 報 告

報名編號：_____

中華民國 九十五年 月 日

注意：

- 1 為維護比賽公平性，請勿於報告中洩露比賽隊伍之身分。
- 2 為方便評比，敬請依此格式撰寫報告，若有項目不適用或未曾完成於您的作品中，該項目請留空白，勿刪除。
- 3 最多請勿超過十五頁，內文字體大小以12點字，行距以1.5行距為原則。
- 4 報告得以中文或英文撰寫。
- 5 請以PDF為檔案格式。

- 1 摘要
- 2 作品簡介
- 3 實作內容描述(設計考量、軟體之創意性及最佳化方法)
- 4 實作結果
- 5 結論
- 6 參考文獻

附件二 初審查意見表

2006【ARM Code-O-Rama】設計大賽初審查意見表

分項總合總分請給 1~5 分：5 分—非常滿意；4 分—滿意；3 分—普通；2 分—不滿意；1 分：非常不滿意
給分可給至小數點一位，例如：3.5 分。

報名編號

題目

軟體實作設計考量。註記：

軟體創意性與最佳化方法。註記：

報告撰寫完整度。註記：

總分：_____分 評審委員簽名：_____日期：_____

附件三 現場報告審查表

2006【ARM Code-O-Rama】設計大賽現場報告審查意見表

分項總合總分請給 1~5 分：5 分—非常滿意；4 分—滿意；3 分—普通；2 分—不滿意；1 分：非常不滿意
給分可給至小數點一位，例如：3.5 分。

報名編號

題目

軟體實作設計考量。註記：

軟體創意性與最佳化方法。註記：

作品完成度（包含軟體效能、Code Size、Cycle count...）。註記：

口頭報告內容解說。註記：

總分：_____分 評審委員簽名：_____日期：_____

附件四 CIC 驗證意見表

2006【ARM Code-O-Rama】設計大賽CIC驗證意見表

分項總合總分請給 1~5 分：5 分—非常滿意；4 分—滿意；3 分—普通；2 分—不滿意；1 分：非常不滿意

給分可給至小數點一位，例如：3.5 分

@ 注意事項：

1. 若軟體功能錯誤即零分。
2. 作品完成度佔80%分數，檔案結構、設計環境設定與說明文件之完整清楚性佔20%分數。

報名編號	
題目	
作品完成度（包含軟體效能、Code Size、Cycle count...）。（80%）註記：	
檔案結構（是否清楚、易懂；檔案是否完整），設計環境說明及設定（是否易於port到驗證者的設計環境），說明文件（Readme.txt）之完整性、清楚性（是否簡潔清晰易懂，包含設計、驗證等說明）。（20%）註記：	

總分：_____分 評審委員簽名：_____日期：_____

附件五 評分用檔案與檔案上傳

參與競賽之隊伍必須將Project建立在“c:\arm_contest\team_xxx”路徑下，其中“xxx”為各隊伍之編號，並將Project name命名為“ljpgtopnm”，而各位同學所要繳交之程式（*.c、*.h、*.s）則必須放置在“c:\arm_contest\team_xxx\code”之路徑下，並在繳交資料時，將整個team_xxx資料夾（內含Project及code）壓縮成RAR格式，並將此壓縮檔命名為update_xx.rar，其中xx為上傳之次數。例如隊伍編號為“123”之隊伍，必須在“c:\arm_contest”路徑下建立一資料夾“team_123”，並將Project建立在“c:\arm_contest\team_123”路徑下；最後必須將整個team_123資料夾壓縮並上傳至FTP，而壓縮檔名則視上傳次數而定，假設第一次上傳此壓縮檔，則壓縮檔名須命名為update_00.rar，而第二次上傳檔名就必須命名為update_01.rar，以此類推，評審將以最後上傳之檔案進行評分作業。下表六為詳細所須繳交資料。

表六 繳交資料

File/Directory	Path	Description
.c、.cpp	c:\arm_contest \team_xxx\code	所有跟 JPEG-LS 解壓縮程式有關之 C 或 C++檔
*.h	c:\arm_contest \team_xxx\code	所有跟 JPEG-LS 解壓縮程式有關之標頭檔
*.s	c:\arm_contest \team_xxx\code	所有跟 JPEG-LS 解壓縮程式有關之組合語言檔
Readme_xxx.txt	c:\arm_contest \team_xxx	程式說明檔，並依照各組編號命名
ljpgtopnm	c:\arm_contest	整個 Project 資料夾，此資料夾須命名為 ljpgtopnm
初賽報告書_xxx.doc	c:\arm_contest	請將附件一的初賽報告書內容填妥，並依照各組編號命名

上傳網址:<http://arm-contst.com>

1. 再次提醒各參賽隊伍，壓縮檔team_xxx.rar、Readme_xxx與初賽報告書_xxx.doc之“xxx”編號需一致！
2. 建議各參賽隊伍，無論是否完成設計，均需上傳設計檔案！

¹ 假如參賽者無任何壓縮軟體，您可以至以下連結，下載壓縮軟體試用版。

winrar: <http://www.rarlab.com/index.htm>