Computer Architecture
CPSC 32, Fall Semester 2003
Project #0
Due: Two weeks after your lab session – complete in teams of 2 or less

1 Assignment

Implement the LZW compression algorithm. LZW is short for Lempel, Ziv, and Welch, the inventors of this compression method. The LZW compression algorithm is a dictionary based lossless compression method that is used, for instance, in the Unix `compress` program. We follow Nelson [1] in our exposition.

Encoding. The encoding algorithm is given by

```
LZW_COMPRESS
STRING = get input character
WHILE there are still input characters DO
    CHARACTER = get input character
    IF STRING+CHARACTER is in the string table then
        STRING = STRING+character
    ELSE
        output the code for STRING
        add STRING+CHARACTER to the string table
        STRING = CHARACTER
    END of IF
END of WHILE
output the code for STRING
```

An example will illustrate the features. Suppose that we want to encode the string `/WED/WE/WEE/WEB/WET`. We assume that the extended ASCII alphabet is already in the code `table[0..255]`. We read the first character `/`, then the second character `W`. Since `/W` is not in the table, the new entry `table[256]=/W` is created[1], and the code for `/` is output.

[1] Do not take this literally, because storing full strings would be inefficient.

character input	output code	new code value	new string
/W	/	256	/W
E	W	257	WE
D	E	258	ED
/	D	259	D/
WE	256	260	/WE
/	E	261	E/
WEE	260	262	/WEE
/W	261	263	E/W
EB	257	264	WEB
/	B	265	B/
WET	260	266	/WET
EOF	T		

Decoding. The decoding procedure is given by

```
LZW_DECOMPRESS
Read OLD_CODE
output OLD_CODE
WHILE there are still input characters DO
    Read NEW_CODE
    STRING = get translation of NEW_CODE
    output STRING
    CHARACTER = first character in STRING
    add OLD_CODE + CHARACTER to the translation table
    OLD_CODE = NEW_CODE
END of WHILE
```

Decoding `/ W E D 256 E 260 261 257 B 260 T` is summarized in the following table.

NEW_CODE	OLD_CODE	output	char	table entry
/	/	/		
W	/	W	W	256 = /W
E	W	E	E	257 = WE
D	E	D	D	258 = ED
256	D	/W	/	259 = D/
E	256	E	E	260 = /WE
260	E	/WE	/	261 = E/
261	260	E/	E	262 = /WEE
257	261	WE	W	263 = E/W
B	257	B	B	264 = WEB
260	B	/WE	/	265 = B/
T	260	T	T	266 = /WET

Your task is to implement the encoding and decoding algorithms of the LZW compression method in MIPS assembly language. The class homepage will contain hints and further instructions that will be frequently updated. In particular, we will provide links to the article by Nelson [1]. This article contains an implementation of the LZW compression method in C. Use this as a guideline for your assembly language program.

Before you start coding the assembly language part, make sure that your thoroughly understand the method. Analyze the C implementation carefully, because the implementations details are nontrivial.

2 Dishonesty

Make sure that you complete the assignment by your team member only. Do not copy the code from others, nor provide others with your code. Refrain from copying and modifying the code from other sources.

References

[1] M. Nelson. *LZW data compression*, Dr. Dobbs Journal, October, 1989.