CSEE 3827: Fundamentals of Computer Systems

Project #3

Huffman Decoder

Due 11/7/17 at 11:59PM

1 Background

Huffman Coding. Huffman codes are a variable-length character code that encodes characters based on their frequency. Frequently seen characters are given short codewords, while rare characters have longer code bits. Overall this tends to compress the data. See Wikipedia for more background on the topic: https://en.wikipedia.org/wiki/Huffman_coding

Our Huffman Tree Decoding Huffman codes entails traversing a binary tree according to the sequence of code word bits until a leaf node containing a character is found. When the code bit is 0, one traverses the left subtree; when it is 1, one traverses the right subtree.

Our Huffman tree is encoded with 17-bit values. The most significant bit indicates whether or not the node is a leaf.

Non-leaves are represented as:

$node_{16}$	0
$node_{15:8}$	address of left child
$node_{7:0}$	address of right child

Leaves are represented as:

$node_{16}$	1
$node_{7:0}$	ASCII character code

For your reference, we have provided both the code table and tree at the end of this document. You will also find the tree encoded, as described in a ROM in the scaffolding. The root of our tree is found at address zero.

2 Function

Your HUFFDEC module should performs this decoding traversal. It accepts a sequence of bits, transferred via the same handshaking protocol as in the last assignment. For each sequence of bits that represents a character your module should output the corresponding 8-bit ASCII character.

[Update 10/23/17: For this assignment, you are free to use any of the components available in Logisim.]

3 Quality

There is no quality score for this assignment.

4 Scaffolding

The test harness works as in the previous assignment. It decodes all 95 characters in random order.

5 Rules and Regulations

- Do not change the name or appearance of the HUFFDEC module.
- Simulation of the test harness must halt without error (e.g., oscillations or undefined wire values). Incorrect results ($NUM\_FAIL > 0$) are fine.
- Submissions must be made via courseworks.
- Upload a single .circ file, [Update 10/30/17: ~~whose names contains only alpha-numeric characters, hyphens, or underscores (no spaces, parenthesis, etc.)~~]. Do not upload lib3827.circ.
- The uploaded file should have no external libraries loaded (i.e., external dependencies) except for lib3827.

6 Scoring Rubric

To be graded, your submussion must adhere to all rules and regulations. We will test your submission by attempting to decode all 95 characters, but in a different random order from the scaffolding.

	Total	Formula
Function	100 pts	If $NUM\_PASS == 0$, decodes nothing, 0 If $NUM\_PASS == 1$, decodes one character only, 40 If $NUM\_PASS > 1$, $40 + \lceil 60 \times \frac{NUM\_PASS}{95} \rceil$
Style	10 pts	If fully functional, extra credit to ten (or more) most beautiful schematics.

7 Our Huffman Code

0010	r
0011	o
0110	}
1001	e
1011	a
00000	h
00001	3
00010	p
01000	c
01001	0
01010	d
01110	m
01111	2
10101	l
11001	t
000110	w
000111	f
010111	y
100000	7
100001	8
100011	6
101000	9
101001	g
110000	5
110001	4
110100	k
0101100	z
1000100	v
1000101	j
01011011	q
11010110	x
0101101000	M
0101101001	T
0101101010	B
1101010000	R
1101010010	E
1101010100	S
1101010111	A
01011010111	F
11010100010	K
11010100111	G
11010101010	H
11010111000	C
11010111001	D
11010111010	I
11010111100	O
11010111101	P
11010111110	N
11010111111	L
010110101101	-
110101000110	V
110101000111	@
110101001100	*
110101010110	Y
110101011000	!
110101011001	.
110101011010	W
110101110110	U
110101110111	J
0101101011001	$
1101010011010	-
1101010101110	X
1101010101111	Q
1101010110111	Z
11010101101101	#
0101101011000010	+
1101010011101101	/
1101010011101110	,
01011010110000000	[
01011010110000001	\
01011010110000010	)
01011010110000011	‘
01011010110000111	&
11010100110110000	=
11010100110110001	~
11010100110111110	%
11010100110111111	space
11010101101100001	^
11010101101100010	;
11010101101100011	?
010110101101100100	:
110101011011000001	]
010110101011000011011	(
110101011011000000	<
010110101100011010 0	{
010110101100011010 1	double quote
110101011011000010	>
1101010110110000010	|
1101010110110000011	’

8 Contents of Tree ROM

Addr	$node_{16}$	$node_{15:8}$	$node_{7:0}$
0:	0	187	188
1:	1		32
2:	1		36
3:	1		40
4:	1		44
5:	1		48
6:	1		52
7:	1		56
8:	1		60
9:	1		64
10:	1		68
11:	1		72
12:	1		76
13:	1		80
14:	1		84
15:	1		88
16:	1		92
17:	1		96
18:	1		100
19:	1		104
20:	1		108
21:	1		112
22:	1		116
23:	1		120
24:	1		124
25:	1		35
26:	1		39
27:	1		43
28:	1		47
29:	1		51
30:	1		55
31:	1		59
32:	1		63
33:	1		67
34:	1		71
35:	1		75
36:	1		79
37:	1		83
38:	1		87
39:	1		91
40:	1		95
41:	1		99
42:	1		103
43:	1		107
44:	1		111
45:	1		115
46:	1		119
47:	1		123
48:	1		34
49:	1		38
50:	1		42
51:	1		46
52:	1		50
53:	1		54
54:	1		58
55:	1		62
56:	1		66
57:	1		70
58:	1		74
59:	1		78
60:	1		82
61:	1		86
62:	1		90
63:	1		94
64:	1		98
65:	1		102
66:	1		106
67:	1		110
68:	1		114
69:	1		118
70:	1		122
71:	1		126
72:	1		33
73:	1		37
74:	1		41
75:	1		45
76:	1		49
77:	1		53
78:	1		57
79:	1		61
80:	1		65
81:	1		69
82:	1		73
83:	1		77
84:	1		81
85:	1		85
86:	1		89
87:	1		93
88:	1		97
89:	1		101
90:	1		105
91:	1		109
92:	1		113
93:	1		117
94:	1		121
95:	1		125
96:	0	24	26
97:	0	47	48
98:	0	55	96
99:	0	97	3
100:	0	8	98
101:	0	54	99
102:	0	100	87
103:	0	39	16
104:	0	74	17
105:	0	101	49
106:	0	79	71
107:	0	73	1
108:	0	102	63
109:	0	31	32
110:	0	103	104
111:	0	27	105
112:	0	106	28
113:	0	4	107
114:	0	108	109
115:	0	110	111
116:	0	112	113
117:	0	114	25
118:	0	115	2
119:	0	40	116
120:	0	15	84
121:	0	117	62
122:	0	118	75
123:	0	61	9
124:	0	50	119
125:	0	86	120
126:	0	72	51
127:	0	38	121
128:	0	85	58
129:	0	122	57
130:	0	35	123
131:	0	124	34
132:	0	11	125
133:	0	126	127
134:	0	33	10
135:	0	82	128
136:	0	36	13
137:	0	59	12
138:	0	83	14
139:	0	56	129
140:	0	60	130
141:	0	81	131
142:	0	37	132
143:	0	133	80
144:	0	134	135
145:	0	136	137
146:	0	138	139
147:	0	140	141
148:	0	142	143
149:	0	144	145
150:	0	146	92
151:	0	147	148
152:	0	23	149
153:	0	70	150
154:	0	69	66
155:	0	151	152
156:	0	46	65
157:	0	153	94
158:	0	30	7
159:	0	154	53
160:	0	78	42
161:	0	77	6
162:	0	43	155
163:	0	93	64
164:	0	19	29
165:	0	21	156
166:	0	41	5
167:	0	18	157
168:	0	91	52
169:	0	158	159
170:	0	160	20
171:	0	161	22
172:	0	162	76
173:	0	163	67
174:	0	45	90
175:	0	164	165
176:	0	68	44
177:	0	166	167
178:	0	95	168
179:	0	169	89
180:	0	170	88
181:	0	171	172
182:	0	173	174
183:	0	175	176
184:	0	177	178
185:	0	179	180
186:	0	181	182
187:	0	183	184
188:	0	185	186