Experiments with the Huffman Decoder/Encoder (VIS bench)

Cécile Braunstein

December 13, 2012

1 Original platform from VIS

1.1 Model architecture

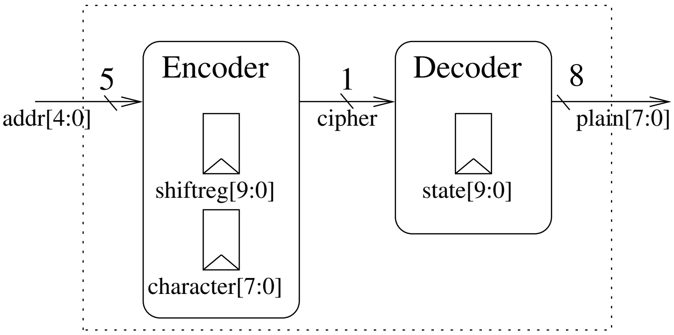
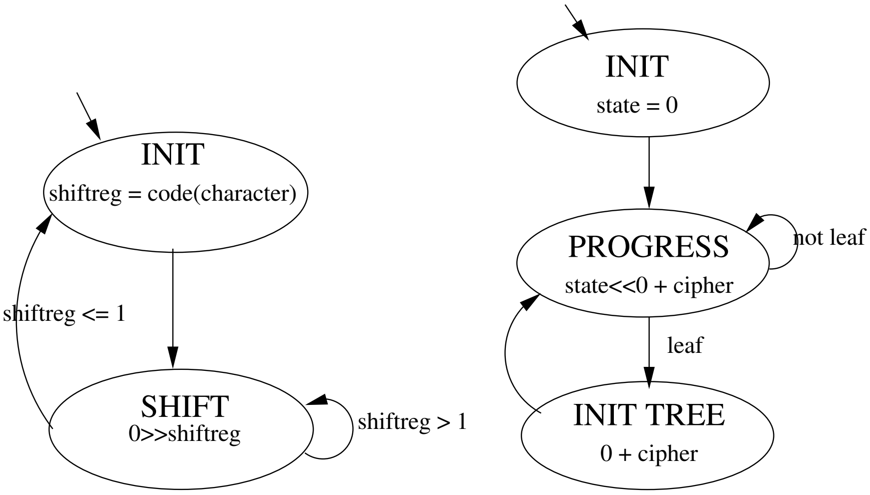


Figure 1: Architecture of Huffman original



(a) Encoder automaton

(b) Decoder automaton

The Huffman circuit describes the implementation of static (English) Huffman encoder/decoder.

- Encoder: Read a character and return the encoding form bit by bit
- Decoder: Explore the encoding tree and return the character when it reaches a leaf.

1.2 Robustness Analysis

The robustness analysis has been performed for usut and msmt faults only. The set of safe states for robustness 4 is the reach set of states. The table 1 shows the robustness analysis of the original circuit in

Rob	**Fault**	$\|Err\|$	ν_{ev}	ν_{pot}	**time**
Rob4	SEU	61810	48%	99.5%	0
	MEU	2.68e+08	7%	99.8%	0
Rob3	SEU	1.53e+06	44.7%	99.6%	1min
	MEU	2.36e+11	0.6%	99.8%	1min
Rob1	SEU	1.43e+06	30.4%	-	0
	MEU	2.36e+11	0%	-	0

Originaly reachable states : 878

Table 1: Robustness of the Huffman circuit from the VIS bench

VIS bench. There exists *eds* from where the circuits will never recover. After analyzing these path (by producing counterexample), we were able to avoid them path by modifying the encoder automaton. The case were `shiftreg` equals 0 is not taken into account on this automaton. This can be solved by replacing the guard transition from SHIFT to INIT.

From this point, we will considered only this modified version of the Huffman circuit.

Rob	**Fault**	$\|Err\|$	ν_{ev}	ν_{pot}	**time**
Rob4	SEU	61654	48%	100%	0
	MEU	2.68e+08	7%	100%	0
Rob3	SEU	1.43e+06	45%	100%	1min
	MEU	2.36e+11	0.6%	100%	1min
Rob1	SEU	1.43e+06	30.5%	-	0
	MEU	2.36e+11	0%	-	0

Originaly reachable states : 878

Table 2: Robustness of the corrected Huffman circuit from the VIS bench

This circuit is not completely robust due to the desynchronization between the encoder and the decoder machine. The fault may have change the code of character in the encoder while the decoder where decoding it. The original circuit does not have a mechanism to prevent such desynchronization such as handshake protocol.

2 Huffman decoder/encoder synchronization

We synchronized the behavior of the decoder and the encoder device. The decoder starts only when the encoder said so. The encoder may restarts the decoder even when the decoding part is not over.

This mechanism ensure the robustness level 4 with the set of reachable states considered as safe states. Nevertheless this does not ensure the synchronization with the golden behavior. The non-robustness 3 comes from the fact that the environment does not have constraints. Moreover, it comes also from the lack of synchronization between the environment and the model under test.

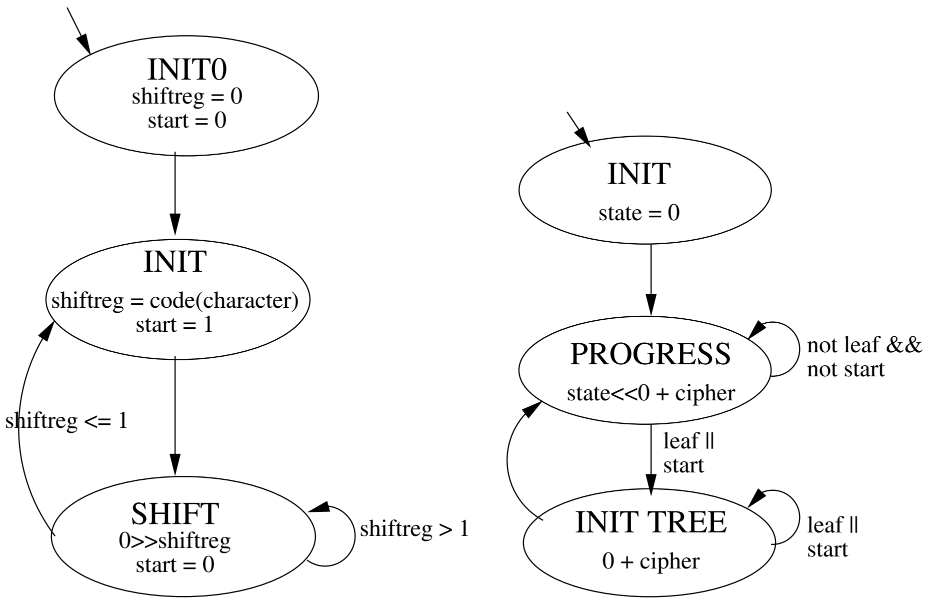
3 Huffman with a constraint environment

3.1 Environment: a given text

We build an environment synchronized with the encoder. The environment send the letter A and E alternatively when it received the acknowledgment from the encoder saying that the character code is sent.

Protected registers	Fault type	Rob 4		Rob 3		Rob 1
		ν_{ev}	ν_{pot}	ν_{ev}	ν_{pot}	$\nu_{ev} = \nu_{pot}$
character	SEU	25.34%	100%	22.78%	100 %	2.72%
	MEU	7.07%	100%	0.59%	100%	0%
state	SEU	58.32%	100%	55.30%	100%	44.61%
	MEU	24.78%	100%	2.22%	100%	0.10%
shiftreg	SEU	62.80%	100%	58.90%	100%	47.30%
	MEU	10.92%	100%	13.86%	100%	0.62%
character/ state	SEU	25.87%	100%	19.54%	100%	0.30%
	MEU	24.78%	100%	2.12%	100%	0%
character/ shiftreg	SEU	24.62%	100%	26.02%	100%	5.14%
	MEU	7.47%	100%	13.79%	100%	0.623%
state/ shiftreg	SEU	100%	100%	100%	100%	100%
	MEU	100%	100%	100%	100%	100%

Table 3: Register grading of the corrected Huffman circuit from the VIS bench



(c) Encoder automaton

(d) Decoder automaton

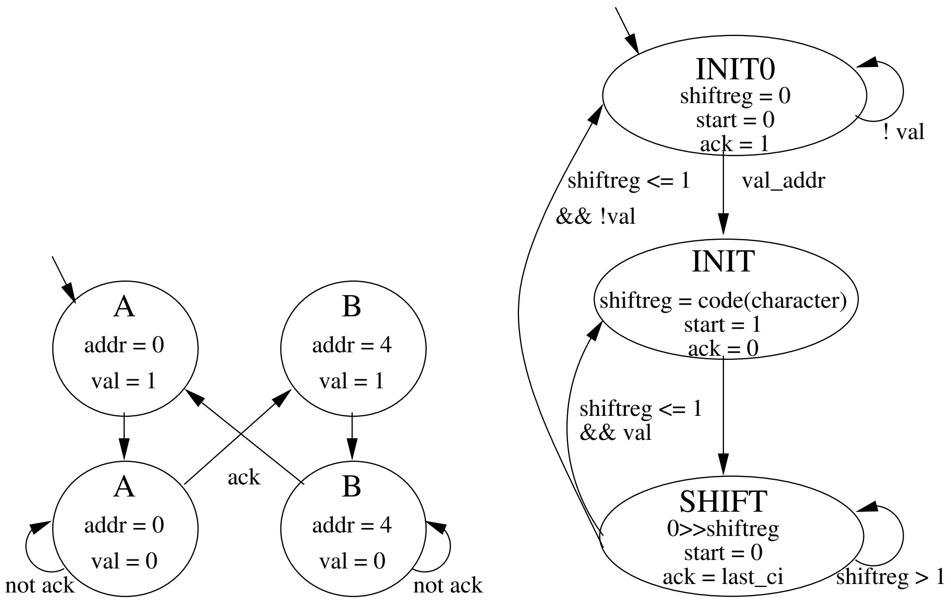
Rob	**Fault**	$\|Err\|$	ν_{ev}	ν_{pot}	**time**
Rob4	SEU	25 633	100%	100%	0
	MEU	5.37e+08	100%	100%	0
Rob3	SEU	334 248	83.5%	100%	0.5
	MEU	4.72e+11	8.5%	100%	0.5
Rob1	SEU	334 248	32.7%	-	0.7
	MEU	4.72e+11	0%	-	2

Originaly reachable states : 879

Table 4: Robustness Huffman circuit with internal synchronization

Protected	Fault	Rob 3		Rob 1
registers	type	ν_{ev}	ν_{pot}	$\nu_{ev} = \nu_{pot}$
character	SEU	77%	100 %	7%
	MEU	8.5%	100%	0%
state	SEU	74.77 %	100%	46.64%
	MEU	8.5%	100%	0%
shiftreg	SEU	100%	100%	49.73%
	MEU	100%	100%	1.96%
start	SEU	82.88%	100%	30.89%
	MEU	100%	100%	0%
character/ state	SEU	56.42%	100%	7.84%
	MEU	8.56%	100%	0%
character/ shiftreg	SEU	100%	100%	13.16%
	MEU	100%	100%	1.96%
state/ shiftreg	SEU	100%	100%	98.12%
	MEU	100%	100%	15%
state/ start	SEU	73.37%	100%	44.62%
	MEU	8.53%	100%	0%
character/ start	SEU	76%	100%	3.25%
	MEU	7.8%	100%	0%
shiftreg/ start	SEU	100%	100%	47.87%
	MEU	100%	100%	3.45%
shiftreg/ state/start	SEU	100%	100%	100%
	MEU	100%	100%	100%

Table 5: Register grading Huffman circuit with internal synchronization



(e) Environment

(f) New Encoder

We perform a robustness analysis with a new rob3 :
The faulty and the golden have got the same inputs from the environment. The environment only reacts to the outputs from the faulty circuits.

The set of safe states are the ones where the golden and the faulty states are observably equivalent.

Rob	**Fault**	$\|Err\|$	ν_{ev}	ν_{pot}	**time**
Rob4	SEU	270	100%	100%	0
	MEU	4.29e+09	100%	100%	0
Rob3	SEU	270	100%	100%	0.5
	MEU	8.27e+10	100%	100%	0.5
Rob1	SEU	270	47.7%	-	0.7
	MEU	8.27e+10	0%	-	2

Originaly reachable states : 90

Self stabilized after 10 steps, there is not healthy states before. Just checking the `AX:9 AG (golendOut == faultyOut)` and `AX:10 AG (golendOut == faultyOut)`

3.2 Synchronized with the decoder

Our goal is to model the basis functionnalities that an environment should at least have. In this experiments any character may be encode. The environment is waiting that the encoder is ready to take a letter (`ack = 1`) and send a letter in a non deterministic time (free input `i_val`). This modeling implies that we need to add fairness constraints to avoid the infinite path where the environment never asks for a new encoding.

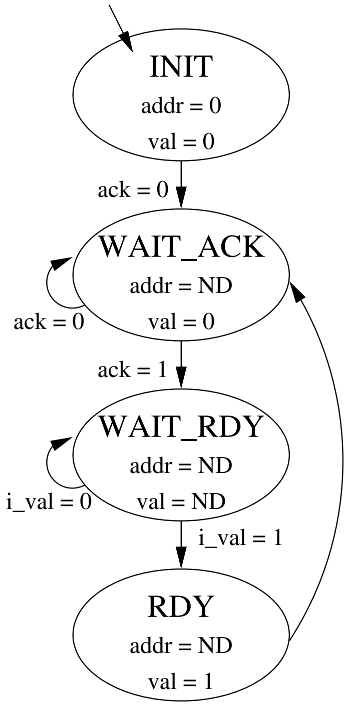


Figure 2: Less constraint environment

3.3 Separate study for decoder and encoder

The encoder is completely robust (rob3 and rob4) under the fairness constraint that the environment should ask for a new computation from time to time.

The decoder has got the same characteristic. It is robust iff his environment always need a new computation. The encoder verifies this property `(AG(AF(start = 1)))`, hence the composition of the decoder and encoder is 100% robust.

Rob	**Fault**	$\|Err\|$	ν_{ev}	ν_{pot}	**time**
Rob4	SEU	6.42e+05	100%	100%	0
	MEU	1.03e+11	100%	100%	0
Rob3	SEU	6.42e+05	100%	100%	7
	MEU	7.32e+13	100%	100%	7
Rob1	SEU	6.42e+05	0%	-	0
	MEU	7.32e+13	0%	-	2

Originaly reachable states : 21 121

Table 6: Robustness with any text environment