World Academy of Science, Engineering and Technology
International Journal of Electrical and Information Engineering
Vol:9, No:2, 2015

Simulation of Hamming Coding and Decoding for Microcontroller Radiation Hardening

Rehab I. Abdul Rahman, Mazhar B. Tayel

***Abstract*—**This paper presents a method of hardening the 8051 micro-controller, able to assure reliable operation in the presence of bit flips caused by radiation. Aiming at avoiding such faults in the 8051 micro-controller, Hamming code protection was used in its SRAM memory and registers. A VHDL code has been used for this hamming code protection.

***Keywords*—**Radiation, hardening, bitflip, hamming code.

I. Introduction

MILITARY, avionics and aerospace applications utilize advanced electronic systems with microprocessors in order to meet spacecraft requirements such as physical volume, weight, power and cost. Fault-tolerance and high-reliability have always been essential attributes of these systems, to keep them operational in such hostile environment [1].

Digital circuits operating in space are subject to different kinds of radiation, which effects can be permanent or transient [2]. The space radiation environment consists of various particles that may interact with digital microelectronic devices causing undesirable effects. Particles of concern are electrons, protons, photons, alpha particles and heavier ions. Permanent effects result of particles trapped at the silicon/oxide interfaces, and they happen only after a certain period of exposure to radiation. On the other hand, transient effects may be provoked by the impact of a single charged particle (Single Event Effects, SEE) in sensitive circuit zones.

According to the impact location, two kinds of SEEs are distinguished: SEUs (Single Event Upsets) and SELs (Single Event Latchups). SEUs are responsible for transient changes in bits of information stored within an integrated circuit. SEUs are a major concern in space environment, and have also been observed on the earth atmosphere as the result of interactions of neutrons [3]. SELs result from the triggering of parasitic transistors presented in CMOS technology, and they provoke short circuits, capable to damage the component by thermal effect if the circuit is not powered-off at time.

The consequences of SEUs depend on the nature of the perturbed information, ranging from erroneous results to system crashes. For complex circuits like DSP processors, coprocessors, or micro-controllers, the sensitivity to SEUs correlates strongly with the amount of internal memory (registers, memory bits, flip-flops, etc).

Fault tolerance techniques attempt to improve reliability by reducing the occurrence of faults. Radiation hardening using specific process technology is an expensive process and, when used for a low-volume production, will lead to very costly parts. The need for low-cost, state-of-the-art high performance computing systems in these areas has been pushing researchers to investigate new fault-tolerance technique applications.

This paper presents a radiation hardened version of the 8051 micro-controller designed with a VHDL description protected by the Hamming Code technique. The choice of this micro-controller was based on the fact that it is widely used in space applications, and consequently, large amount of data about its behavior under radiation is available. Besides, its hardening makes possible the reuse and protection of all systems that are already running based on it. The SEU sensitive area of the microprocessor has been protected using the Hamming code technique. This technique is mainly based on the inclusion of error detecting and correcting capabilities [2].

The paper is organized as follows: Section II presents radiation hardening of a microcontroller, Section III presents the hamming code technique in registers, Section IV shows the hamming code for EDAC (Error Detection And Correction), Section V shows simulation results of hamming coding and decoding and Section VI presents the summary of the paper.

II. Radiation Hardening a Microcontroller by EDAC

Micro-controllers operating in space environment can be affected by SEU. Thus, memory cells and registers included in microprocessors must be protected to avoid potential transient errors. A previous study of radiation effects in the 8051 architecture has been done in [4], to detect the most vulnerable micro-controller units. Experimental results based on fault injection confirmed that transient faults (SEU) affect mainly the internal RAM memory. For 12000 random injected errors, around 10000 errors have occurred in the internal memory. Concerning this amount of errors, 49.96% were tolerated errors, 47.24% resulted in errors and 2.8% resulted in loss of sequence. The technique used in this work to detect and correct faults in the 8051 memory cells (registers or RAM cells) consists of codifying each memory element by Hamming code [5], and to perform the verification of the stored word every time this memory point is accessed. The advantages of this technique are high reliability circuit and flexible VHDL implementation for fast prototyping.

Rehab I. Abdul Rahman is with the Electrical Engineering Department, University of Alexandria, Egypt (phone: 00201002650580; e-mail: rehab_abdel_rahman@ yahoo.com).

Mazhar B. Tayel is with the Electrical Engineering Department , University of Alexandria, Egypt (phone: 00201116554777; e-mail: profbasyouni@gmail.com).

Fig. 1 shows the schematic diagram of the 8051 like microcontroller.

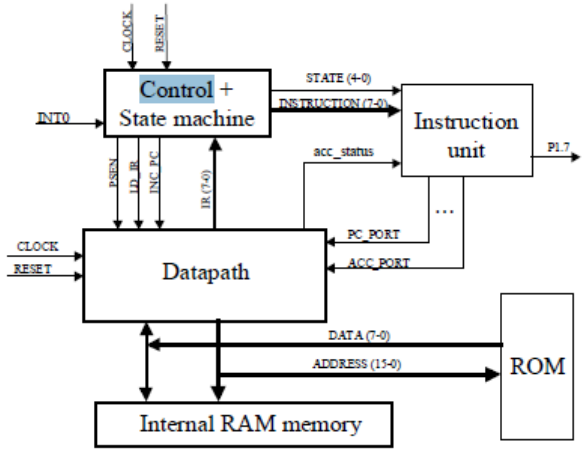

Fig. 1 General Schematic diagram of the available 8051 microcontroller [1]

III. Hamming Code Technique in Registers

Hamming code is an error-detecting and error correcting binary code that can detect all single and double-bit errors and correct all single-bit errors on an n-bit word [6]. A Hamming code satisfies $2^m >= n +1$, where n is the total number of bits in the protected data and m is the number of check bits in the data. This coding method is recommended for systems with low probabilities of multiple errors in a single data structure (e.g., only a single bit error in a byte of data). For example, for an 8-bit data, it is necessary 4 check bits resulting in a new 12-bit data. Fig. 2 shows the main structure used to protect the registers. The 12-bit register in the new design replaces the original 8 bit register.

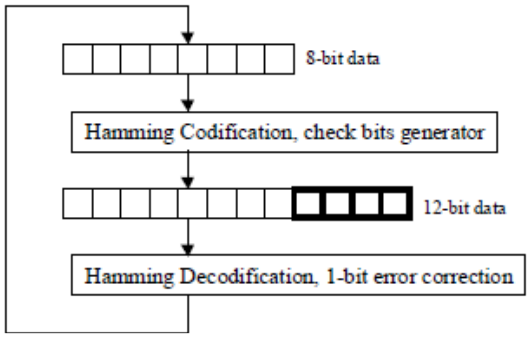

Fig. 2 Hamming Code Technique [1]

IV. Hamming Code for EDAC of Microcontroller

Four structures in the 8051 VHDL description have been protected by Hamming Code: the Finite State Machine, the RAM memory, the Control unit and the Datapath. These blocks contain memory cells that can be affected by SEU radiation effects. The Instruction unit is a completely combinational structure and in principle it is not subject to bit flips, making no necessaries protection schemes for it. The ROM block is also immune to these faults, since its content cannot be modified. VHDL simulations have been used to check the correct Hamming code implementation.

In order to implement the Hamming code, eight combinational components were described in VHDL and used as a package in the 8051 VHDL description. The first component receives an 8-bit data and returns a 12-bit coded word. The second component receives a 12-bit word and returns an 8-bit decoded and corrected data. The size of each Hamming code/decode block grows proportional to the increase of the protected data bits.

A. Transient Fault Detection and Correction

1. In Finite State Machine and Control Unit

The Finite Sate Machine (FSM) and the Control unit represent together only 3% of the sensitive area of the 8051 micro-controller. Although the register sensitive area in these units is much smaller than the internal RAM memory, the effects of a bit flip in this register can be very hazardous for the system. The finite state machine (FSM) of 8051 has 24 states, which have been represented by a 5-bit register to form the next state of the machine. This register must be protected by Hamming code to avoid errors under radiation. Its structure shown in Fig. 3 is similar to the packages previously described.

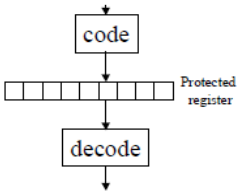

Fig. 3 Hardened registers scheme from the control unit and state machine [1]

In the new FSM, the next state is coded before being stored, and decoded before being used in the control unit. In case of occurrence of a transient fault, the error is corrected.

Since coding and decoding operations are completely combinational, there is no difference in terms of machine cycles needed for an operation.

2. In SRAM Memory

Instances of code blocks with 8-bit input and 12-bit output and decode blocks with 12-bit input and 8-bit output were used to detect and correct transient faults in the SRAM 8051 memory. In the memory access protection, all data words are coded before being stored in the RAM memory, and all memory words are decoded (and possibly corrected) before being used in an operation. Fig. 4 shows this structure.

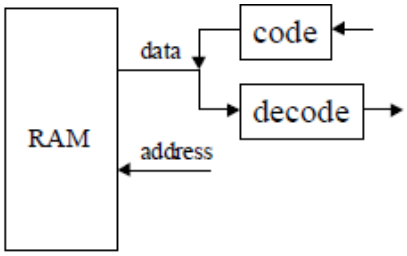

Fig. 4 SRAM memory protected scheme [1]

3. In Datapath

The Datapath is composed of many registers, multiplexors, glue logic and an Arithmetic Logic Unit (ALU) that are enabled according to the signals from the Instruction unit. The clock synchronizes all the registers. The main basic structure is composed of a multiplexor and a register. The same component models for coding and decoding used in the memory were also used to protect the registers in the datapath. There are two version of the Datapath, in the first one, two registers have been protected, the accumulator and the program counter. In the second one, all the registers are protected. Fig. 5 shows the structure used to protect the registers.

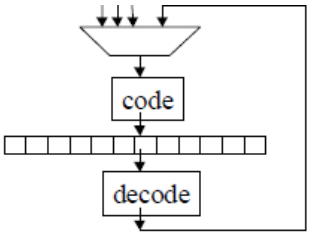

Fig. 5 Hardened Datapath registers scheme [1]

V. Simulation of Hamming Coding and Decoding

For example we are going to consider the coding and decoding of 7bits using the Xilinx software.

The following is the flowchart of hamming codification of a 7 bit word as shown in Fig. 6 and decodification as shown in Fig. 7.

n=number of data bits=7
r=number of redundant bits($2^r \geq n+r+1$)=4
r_1=first redundant bit
r_2=second redundant bit
r_3=third redundant bit
r_4=fourth redundant bit
a=codedword[a(0)a(1)a(2)a(3)a(4)a(5)a(6)a(7)a(8)a(9)
a(10)]
b=data=decoded output[b(0)b(1)b(2)b(3)b(4)b(5)b(6)]

A. Coding Flowchart

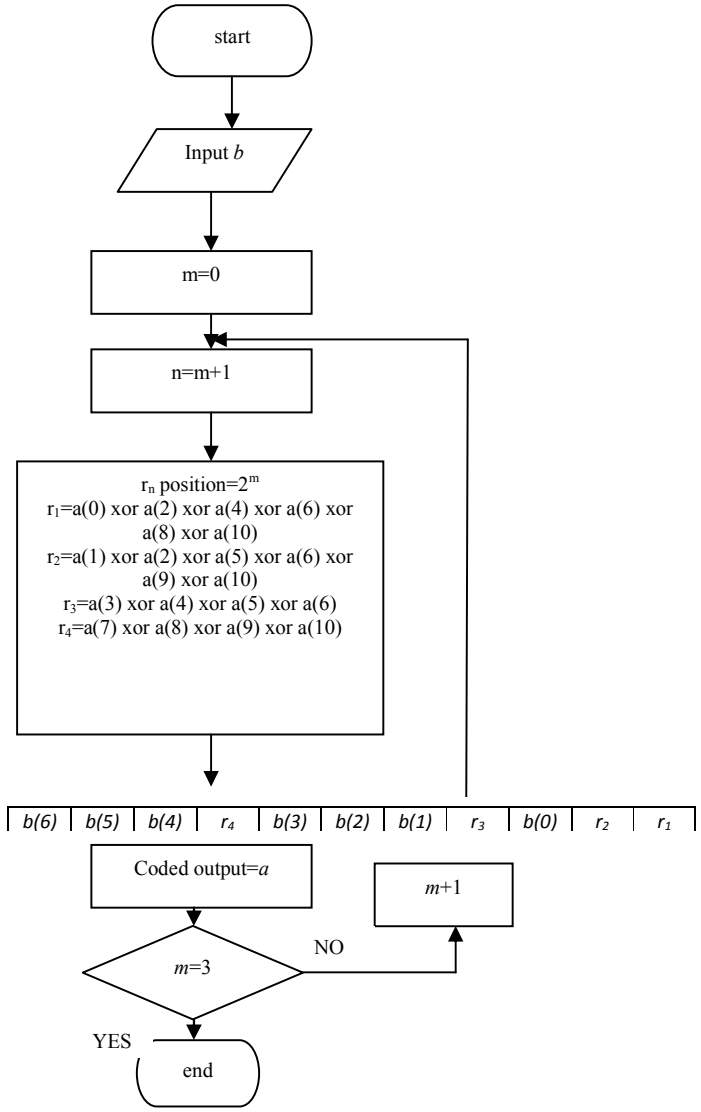

Fig. 6 Flowchart of hamming codification of a 7 bit word b giving a codeword a

B. Decoding Flowchart:

C=[c(0)c(1)c(2)c(3)c(4)c(5)c(6)c(7)c(8)c(9)c(10)]
Binary of R=$r_1r_2r_3r_4$
D=mask

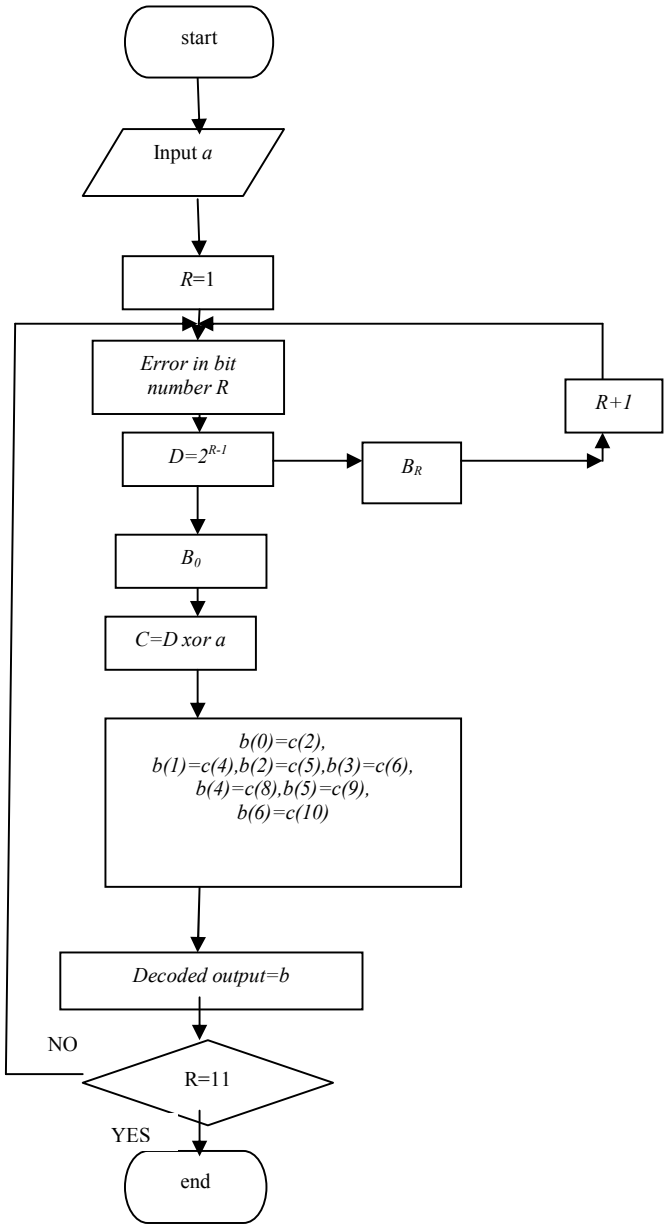

Fig. 7 Flowchart of hamming decodification

The following is the VHDL code for hamming coding of a 7bit word:

```
entity hamcode1 is
   Port ( data_in : in  STD_LOGIC_VECTOR ( 6 downto 0 );
          codata : out  STD_LOGIC_VECTOR ( 10 downto 0 ));
end hamcode1;
architecture Behavioral of hamcode1 is
signal data : std_logic_vector ( 6 downto 0 );
signal coded : std_logic_vector ( 10 downto 0 );
begin
data<=data_in;
process(data)
begin
coded(10)<=data(6);
coded(9)<=data(5);
coded(8)<=data(4);
coded(7)<=data(4) xor data(5) xor data(6);
coded(6)<=data(3);
coded(5)<=data(2);
coded(4)<=data(1);
coded(3)<=data(1) xor data(2) xor d ata(3);
coded(2)<=data(0);
coded(1)<=data(0) xor data(2) xor data(3)xor data(5)xor data(6);
coded(0)<= data(0) xor data(1) xor data(3) xor data(5) xor data(7);
end process;
codata<=coded;
end behavioral;
```

Fig. 8 shows the simulation of hamming coding, for coding "data_in" (1110001). The code word was "coded" (11110000111) .

The following is the VHDL code of hamming decoding:

```
entity hamcodes is
 Port ( data_in : in  STD_LOGIC_VECTOR ( 10 downto 0 );
        decoded : out  STD_LOGIC_VECTOR ( 6 downto 0 ));
end hamcodes;
architecture Behavioral of hamcodes is
signal coded : std_logic_vector ( 10 downto 0 );
signal codata : std_logic_vector ( 10 downto 0 );
 signal codigo : std_logic_vector ( 3 downto 0 );
 signal code : std_logic_vector ( 10 downto 0 );
 signal mask : std_logic_vector ( 10 downto 0 );
 signal decoded_out : std_logic_vector ( 6 downto 0 );
begin
codata<=data_in;
codigo(0)<=codata(0) xor codata(2) xor codata(4) xor codata(6) xor codata(8) xor codata(10) ;
codigo(1)<=codata(1) xor codata(2) xor codata(5) xor codata(6) xor codata(9) xor codata(10);
codigo(2)<=codata(3) xor codata(4) xor codata(5) xor codata(6);
codigo(3)<=codata(7) xor codata(8) xor codata(9) xor codata(10);
process(codigo)
begin
case codigo is
when B"0001"=>mask<=B"00000000001";
when B"0010"=>mask<=B"00000000010";
when B"0011"=>mask<=B"00000000100";
when B"0100"=>mask<=B"00000001000";
when B"0101"=>mask<=B"00000010000";
when B"0110"=>mask<=B"00000100000";
when B"0111"=>mask<=B"00001000000";
when B"1000"=>mask<=B"00010000000";
when B"1001"=>mask<=B"00100000000";
when B"1010"=>mask<=B"01000000000";
when B"1011"=>mask<=B"10000000000";
when others=>mask<=B"00000000000";
end case;
end process;
code<=codata xor mask;
decoded_out(0)<=code(2);
decoded_out(1)<=code(4);
decoded_out(2)<=code(5);
decoded_out(3)<=code(6);
decoded_out(4)<=code(8);
decoded_out(5)<=code(9);
decoded_out(6)<=code(10);
decoded<=decoded_out;
end Behavioral;
```

We assumed that there was an error in the seventh bit such that the input to the decoder became data_in (11111000111), instead of (11110000111). The decoder corrected the error in that bit and its output was as the original input decoded_out=(1110001). Fig. 9 shows the output of the decoder.

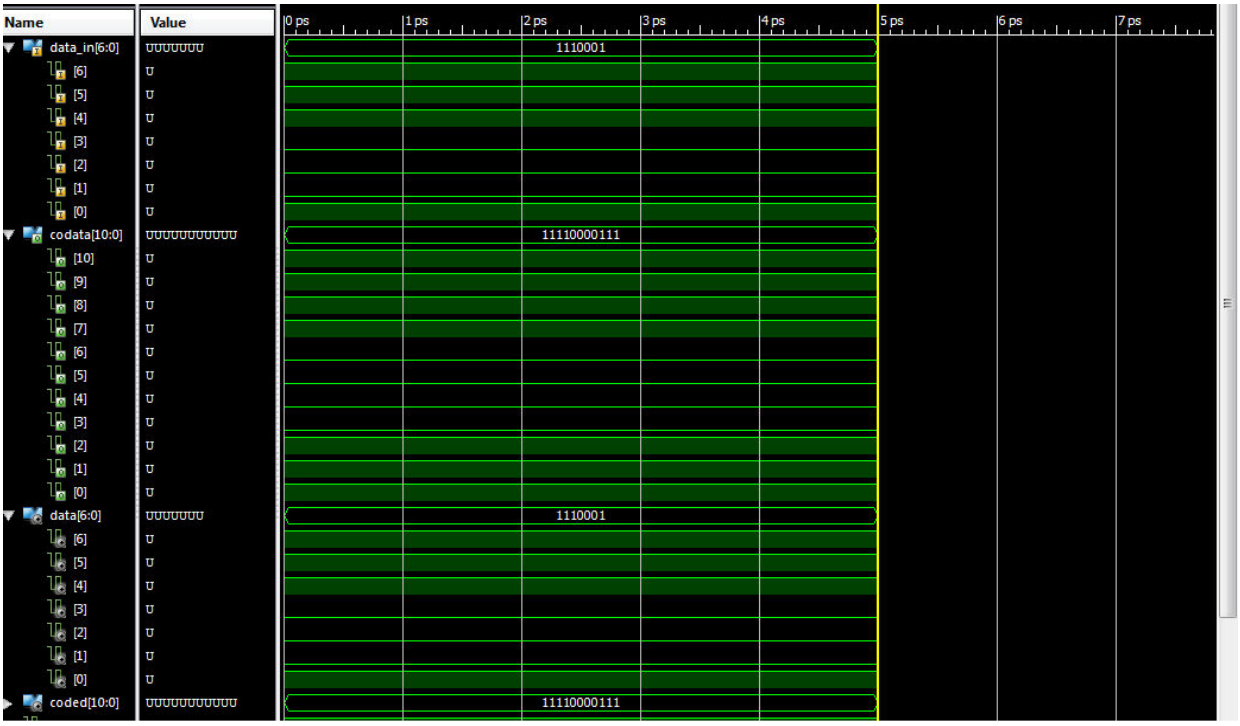

Fig. 8 The simulation of hamming coding, for coding "data_in "(1110001)

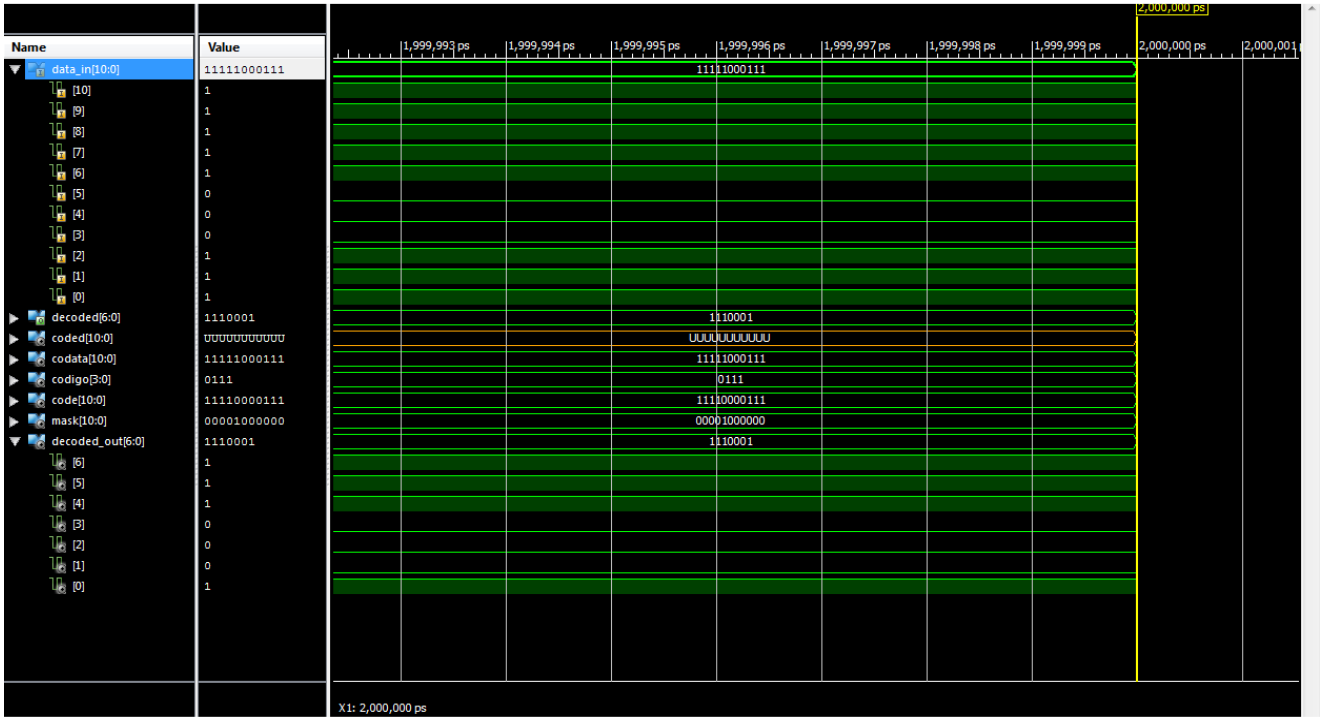

Fig. 9 The output of the decoder

VI. Conclusions

This paper has presented a method of radiation hardening a microcontroller by error detection and correction EDAC. The technique of EDAC used was hamming code. We gave an example for coding a 7bit word using vhdl giving an 11 bit codeword then decoding it returning the 7 bit word.

References

[1] http://ebookbrowsee.net/lima-mapld00-pdf-d42419909
[2] Ma, T.; Dressendorfer, P. Ionizing Radiation Effects in MOS Devices and Circuits, Wiley Eds, New York, 1989.
[3] Normand, E. Single Event Upsets at Ground level.In: IEEE Transactions on Nuclear Science, vol. 43, no.6, Dec, 1996.
[4] Cota, E.; Carro, L.; Lubaszewski, M.;Velazco, R.; Rezgui, S. Synthesis of 8051-like Microcontroller Tolerant to Transient Faults. In: 1st IEEE Latin America Test Workshop (LATW), Brazil,2000.21
[5] Velazco, R; Rezgui, S.; Cheynet, Ph.; Bofill,A.; Ecoffet, R. Thesic: A testbed suitable for the qualification of integrated circuits devoted to operate in harsh environment, IEEE European Test Workshop (ETW'98) pp. 89-90, 27-29 Mai 1998, Spain.26
[6] Kohavi, Z. Switching and Finite Automata Theory, McGraw-Hill, 1970.25