STATE ASSIGNMENT ALGORITHM AND ITS USE IN ELECTRIC VEHICLE'S ECU

Ankit Sharma[1], Narayan Kamath[2], Dr. R.K. Sharma[3],
Department-of-Electronics-&-Communication-Engineering
[1]PG Scholar, Embedded System Design
[2]Senior Solution Architect, KPIT Technologies Bangalore, Karnataka
[3]Professor, School of VLSI and Embedded System Design NIT Kurukshetra, Haryana
India

Abstract— **This paper focus on the issue of electric energy dissipation in code execution on microprocessors of electric vehicles, and proposing use of State Assignment algorithms to minimize energy consumption in vehicle's Electric Control Units. State Assignment Algorithms are used to minimize bit transition at each state change of FSM (Finite State Machine) that provide solution to power dissipation problem in FSM synthesis. We propose use of State Assignment Algorithm in Embedded code to reduce the bit transition that leads to lower energy consumption in CMOS circuit. The primary goal of using State Assignment Algorithm in embedded coding is to reduce switching activity in the state register. In this paper different proposed State Assignment Algorithms will be studied to understand how a state assignment algorithm reduces switching activity. We will go through the basic of how switching activity is related to energy consumption. And also propose a State Assignment Algorithm.**

Index Terms— **ECU, State Assignment Algorithm, FSM, Embedded System, Electric Vehicle.**

I. Introduction

Future of transportation is electric vehicles and number of ECUs is already increasing to make vehicle more user-friendly, smart and safe. It is also making vehicle more reliable and easy for maintenance. There is a lot of computing in vehicle's Electric Control Unit which consumes a significant amount of energy, and when vehicles operations (Transmission and electric control system) completely depends on a battery then it is important to reduce power consumption in Electric control unit. Various low power algorithms may finds Electric vehicles Electronic Control Systems its potential application, to reduce power consumption in Vehicles electronics during code execution. There is lot of research activities to design less power hungry circuit that consumes comparatively less power. There are several approaches to design low power circuit, applicable on different level as low level approaches like voltage scaling and process optimization and high level approaches like instruction set designs and hardware/software co-design [1].

Energy consumption in CMOS circuits can be classified as Dynamic Power and Static Power consumptions. Dynamic Power is main concern the power that is consumed by microprocessor. Dynamic power is based on transition of input bits rather than Static Input. Though energy is consumed by hardware (Microprocessor) but software that is running on hardware chip plays a significant role in controlling the energy consumption in hardware.

If we consider vehicles run by bio-fuel we are not much concerned about code which consumed less power in ECU's during execution time obviously these vehicle have enough electrical power generated by bio-fuel that also gives power to transmission of vehicle.

TABLE I. Different Levels of Embedded System

Different Levels	
Application	→ Algorithm implementation
Operating System	→ Resource Manager
Hardware	→ Actual Energy Consumer

Our purpose is to reduce bit transition in the state register, often power dissipation can be reduced by using encoding in the SIS generated circuits [1].

L. Benini and G.D. Micheli propose a state assignment algorithm that reduces the bit transition per state formulating a general theoretical framework, proposed algorithm trading off computational efforts for quality [2]. K. Kajstura and Dariusz proposed a state assignment decomposition algorithm named as LPBTE (Low Power Binary Tree Encoding) aimed at minimizing power consumption. The proposed algorithm is based on creating a binary tree [3]. Tiago, Lester, Duarte and Osamu presented algorithm that assign codes to every state for Extended Burst-Mode Asynchronous Finite State Machines (XBM_AFSM) that algorithm of State Assignment based on genetic algorithm. XBM_AFSM algorithm can detect the conflicts in XBM specification and insert the minimum number of state variables in the XBM specification so that related conflict can be eliminated [4]. A semi-Gray encoding technique is presented by C. Chen, J. Zhao and Majid Ahmadi that assign codes to every state so that bit transition activity minimizes for state register. A counter with Gray encoding is a proper example which guarantees that the transition between two successive state results in one bit transition [5]. Winfried N¨oth and Reiner Kolla study the issue of providing code to each state for synchronous finite state machines and provide a solution using Spanning Tree Based State Encoding for the purpose of low power dissipation [1].

The paper contains following sections. Section II contains Basics of Power Dissipation in CMOS. Section III describes how A State Assignment Algorithm reduces bit transition in state and energy dissipation in a counter, and how we can use this algorithm in Electric vehicle to reduce electric energy consumption in ECU. Section IV contains one proposed heuristic State Assignment Algorithm. Section V is conclusion.

II. BASIC OF ENERGY DISSIPATION IN CMOS CIRCUIT

Mostly Power dissipation in CMOS circuits can be categories into two categories as Static and Dynamic Power Consumption. Static power consumption occurs due to leakage current in study state. In ideal condition, no static power dissipates in CMOS circuits but in practical scenario the MOS transistor does not behave as an ideal switch. Thus there will always be leakage currents that will give rise to static component of CMOS power consumption. The static power consumption in low-power CMOS circuits should be as low as possible to do that a number of different approaches used to apply at device and circuit level during fabrication of integrated circuit. Thus primarily dynamic Power dissipation will be in focus because it has a significant part in total power dissipation in CMOS circuit. Dynamic power is a result of charging and discharging of load capacitor during logic state transition. During the input voltage (Vin) going high to low or vice versa at some point both the NMOS and PMOS devices will be in on condition as described below.

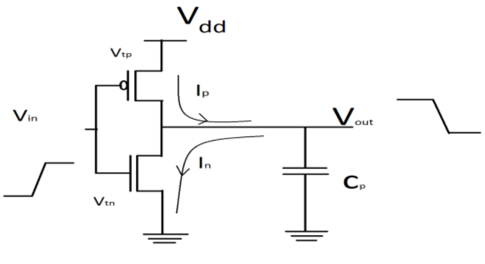

Fig.1. Charging and discharging in CMOS inverter.

In Fig.1 V_{in} is Input Voltage, V_{dd} is Voltage supply, V_{tp}, V_{tn} is threshold voltage for PMOS and NMOS respectively. Ip and In is charging current through PMOS and discharging current through NMOS respectively. Cp parasitic capacitance of CMOS circuit and V_{out} is output voltage. During V_{in} voltage value between V_{tn} and V_{dd} -$|V_{tp}|$, both NMOS and PMOS turn on and a short circuit exists between V_{dd} and ground and to allow current flow. In one full cycle of operation the input voltage going from V_{dd} to ground and return back to V_{dd} again. During V_{dd} to Ground, Pull-down network will cut off and Pull-up network will activate and capacitor starts charging through pull-up network to V_{dd}. In this charging process total energy drawn was $C(V_{dd})^2$ from supply voltage. Half of drawn energy dissipated in the pull-up network (PMOS transistors) and other half stored in load capacitance. During input signal return from ground to V_{dd} Pull-up network go off and Pull-down network (NMOS transistors) activated now capacitor discharge and stored energy dissipate through Pull-Down circuit. The average total power dissipated during one clock cycle (where T is one clock cycle period) is given by [2].

$$P_d = \frac{(V_{dd})^2}{2T} \sum_{i=1}^{n} C_i P_i \qquad (1)$$

where V_{dd} – supply voltage, C_i - load capacitance at the gate i, n – number of gates in the system, P_i – switching probability at the gate i. Power consumption in CMOS circuit depends on the number of bit transition per state or switching activity. To minimize power losses in CMOS circuits is to minimize the number of bit transition per state. The expected number of zero to one logic transition per data cycle (α) coupled with average data rate (f) i.e. αf is directly affect the dynamic power dissipation in CMOS circuit.

$$P_{dyn} = \alpha C(V_{dd})^2 f \qquad (2)$$

III. STATE ASSIGNMENT ALGORITHM AND DYNAMIC ENERGY DISSIPATION

In this section explanation is given how state assignment algorithms help in reducing Power dissipation in CMOS circuit. For an example take n=4 bit counter with 2n number of states. To reduce transition activity every state must have a code those have minimum hamming distance subsequent states in other words minimum transition activity.

Let’s compare two state assignments first one is conventional and second is gray coded state assignment. And observe which state assignment gives minimum bit transition or switching activity. Total bit transition in normal counter is 14 and in Grey code encoded counter is 8.

Suppose every bit transition consume dynamic energy Ed. Then in normal counter consumed energy will be 14Ed and in Grey code encoded state counter it is 8Ed only. Grey code encoded state assignment offers an ideal minimum bit transition (1bit) per state.

TABLE II. NORMAL COUNTER STATES

STATE	**BIT 2**	**BIT 1**	**BIT 0**
STATE 1	0	0	0
STATE 2	0	0	1
STATE 3	0	1	0
STATE 4	0	1	1
STATE 5	1	0	0
STATE 6	1	0	1
STATE 7	1	1	0
STATE 8	1	1	1
SWITCHING	2	4	8

TABLE III . GREY CODE ENCODED COUNTER STATES

STATE	BIT 2	BIT 1	BIT 0
STATE 1	0	0	0
STATE 2	0	0	1
STATE 3	0	1	1
STATE 4	0	1	0
STATE 5	1	1	0
STATE 6	1	1	1
STATE 7	1	0	1
STATE 8	1	0	0
SWITCHING	2	2	4

To State Assignment algorithm can be introduced in various operations in vehicle in order to reduce dynamic power consumption during code execution process in ECU. In general vehicle system takes inputs from various sensors and their reading are depends on vehicle's surrounding and users input (i.e. Breaking and Acceleration paddle, power steering etc.). Reading of sensors in terms of analog voltage goes to ECU, which is converted into digital (binary) value and according to that value processing is done in ECU and commands are send to actuators. We can take one example of electrically controlled breaking system in vehicle, initially without state assignment algorithm then with state assignment algorithm and compare both.

For example let's consider sensor attached to break paddle read break position and send voltage reading according to it to ECU. When there is no break applied sensor read Zero voltage and for full break application sensor read Five volt. An embedded code developer breaks this Zero to Five voltage reading in six different stages (Zero to Five including both reading) and converts these analog voltages to digital values those will be store and updated in a memory register according to sensor reading. According to this binary value processing will be done in ECU and send commands to breaking system that guide how much break will be applied accordingly that represent a task for breaking system. Without state assignment algorithm memory register value (Bit 2, Bit 1, Bit 0) have 10 bit transition when sensor read Zero to Five volt.

Now embedded code developer breaks this Zero to Five voltage reading in six different stages (Zero to Five including both reading) and converts these analog voltages to digital values with State assignment algorithm those will be store and updated in a memory register according to sensor reading. Comparing both method of converting sensor's analog voltage in digital form, second method have 2 bit transition lesser than first method.

IV. PROPOSED STATE ASSIGNMENT ALGORITHM

Proposed state assignment algorithm based on probability of a (Non LSB) bit to be toggled after a toggle of LSB bit in gray code and state condition of blocking bits. In this algorithm extra bits are assigned to every bit except LSB and MSB bits named as blocking bits. The reason behind naming them blocking bit is drawn from the purpose they serve. A Blocking bit assigned to a bit will prevent it to be toggled as long as blocking bit's state is 1. Algorithm will work on basis of five rules introduces by this paper.

TABLE IV. GREY CODE ENCODED STATE ASSIGNMENT

STATE	BIT 3	BIT 2	BIT 1	BIT 0 (LSB)
STATE 1	0	0	0	0 ↷
STATE 2	0	0	0 ↷	1
STATE 3	0	0	1	1 ↷
STATE 4	0	0 ↷	1	0
STATE 5	0	1	1	0 ↷
STATE 6	0	1	1 ↷	1
STATE 7	0	1	0	1 ↷
STATE 8	0 ↷	1	0	0
STATE 9	1	1	0	0 ↷
STATE 10	1	1	0 ↷	1
STATE 11	1	1	1	1 ↷
STATE 12	1	1 ↷	1	0
STATE 13	1	0	1	0 ↷
STATE 14	1	0	1 ↷	1
STATE 15	1	0	0	1 ↷
STATE 16	1 ↷	0	0	0

↷ Represents Bit transition in every bit column.

In algorithm every state is been given a code that results in a minimum bit transition. Proposed algorithm decides which bit to be toggled in next state in order to minimize bit transition. Algorithm describes as bellow with an example of a 4 bit counter while starting state is 0000 (State 1). Since Grey code encoded state assignment gives minimum switching activity, we will create an algorithm to assign codes to each state in order to achieve minimum bit transition.

In algorithm number of bits has been split in to two categories one is LSB bit (Bit 0) and second is Non-LSB Bits

(All bits except Bit 0). Algorithm will give some general rules that decide which bit must be toggle. In any given state a non LSB bit have been toggled then in next state only LSB bit will be toggled. Now if in any given state LSB bit is toggled then algorithm should be able to identify the Bit which needs to be toggled in next state. If we put Bits in rows and states in columns a tree like structure help to define some rules those helps in deciding bit need to be toggle. In Fig.2 states starts with S2 though initial state is S1 but first toggle occurs in S2 state (refer Table III). By observing Fig.2 we can generalize some rules those algorithm will follow during bit assignment.

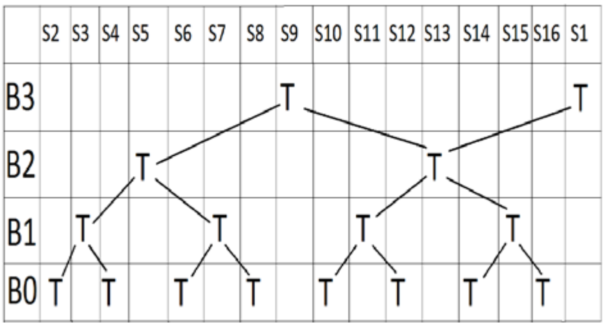

Fig. 2. A tree like structure where 'T' represents bit toggle, B0, B1, B2, B3 are Bits and S represents different states of counter.

Rule 1: If in any given state toggled bit is a Non-LSB bit then in next state LSB bit must be toggled.

Now once LSB bit is toggled in any given state, algorithm need to decide which bit will be toggle in next state. To decide this if we look into Fig.2, after every toggle of bit0 in next state toggled bit was bit1 in 50% of the cases, and bit2 in 25% of the cases.

If we represent every bit with their weight (w_i), then probability of toggling i^{th} bit in next state after LSB bit toggle will be (except LSB bit).

$$P_i = \left(\frac{1}{2^{w_i}}\right) \quad (3)$$

Rule 2: If in given state LSB bit is toggled then in next state Bit will be chosen to toggle according to equation 3. The bit that has higher probability will be given priority to toggle only if it is not blocked to toggle.

Rule 3: If any bit is been blocked to toggle algorithm moved to next higher weighted bit to toggle it.

Rule 4: As a particular weighted bit toggles it blocks itself to toggle in further states and also unblock all lower weighted bits to toggle in further states.

Rule 5: LSB and MSB bits do not block themselves to toggle.

Below a figure is presented that explains working algorithms by following above rules.

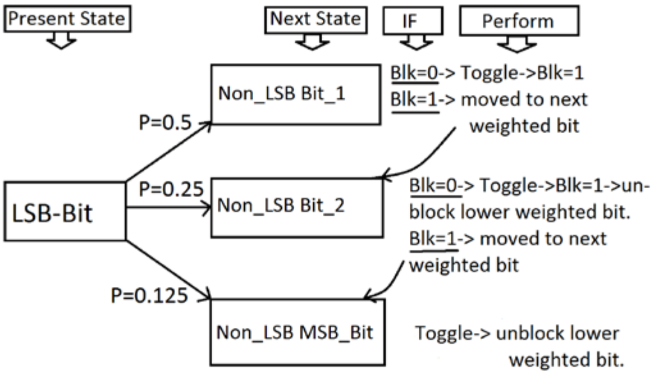
Present State
Next State
IF
Perform
LSB-Bit
P=0.5
P=0.25
P=0.125
Non_LSB Bit_1
Non_LSB Bit_2
Non_LSB MSB_Bit
Blk=0-> Toggle->Blk=1
Blk=1-> moved to next weighted bit
Blk=0-> Toggle->Blk=1->un-block lower weighted bit.
Blk=1-> moved to next weighted bit
Toggle-> unblock lower weighted bit.

Fig. 3. Proposed Algorithm's description.

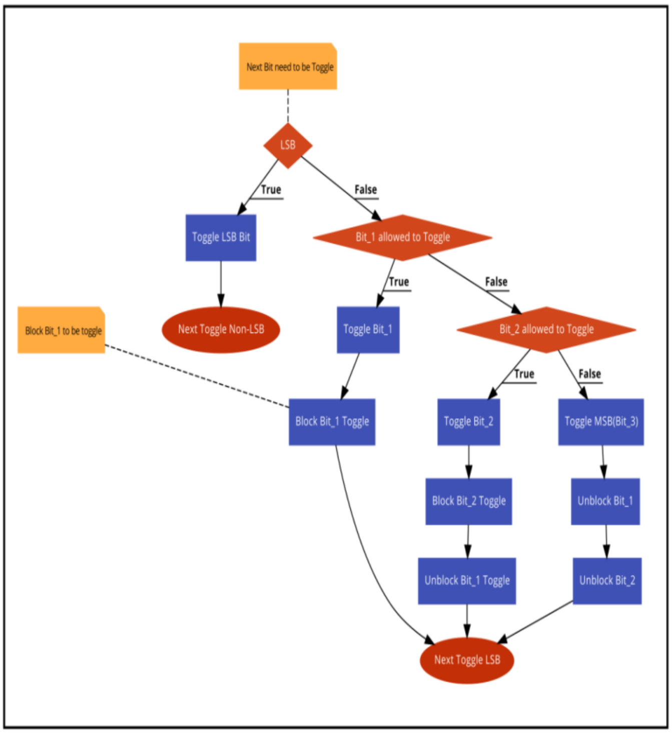

Fig.4. Proposed Algorithm's Flow Diagram.

Algorithm for state assignment is described by Fig. 3 above. In Fig. 3 'Blk' represent blocking bit, every bit except LSB and MSB bits have their own 'Blk' bit.

When Blk bit is logic 0 means bit related to this Blk (Blocking) bit can be toggled. And if Blk bit is logic 1 means bit related to this Blk bit is blocked to be toggled.

For bit selection, algorithm checks probability of every bit to be toggled and bit that has highest probability will be given priority. Algorithm checks status of 'Blk' bit and acts accordingly.

In figure 3 at present state LSB bit is toggled, then in next state algorithm goes to Non_LSB Bit_1 that has highest probability to be toggled and check its Blk bit status. If Blk bit value is logic 0 then algorithm toggle Non_LSB Bit_1 and change Blk bit value to logic 1. But if Blk bit value is already logic 1 then algorithm go to next higher weighted bit (Non_LSB Bit_2) and check its Blk Bit status. If Blk Bit value is logic 0 then algorithm toggle the Non_LSB Bit_2 and change its Blk bit value to logic 1and Blk bit related to Non_LSB Bit_1 to logic 0. If algorithm found Blk bit value related to Non_LSB Bit_2 is logic 1 then algorithm goes to MSB bit to toggle its value and also change value of all Blk bits related to lower weighted bits (Non_LSB Bit_2 and Non_LSB Bit_1) to logic 0. LSB bit and MSB bit does not have Blk bit.

After toggling of any Non_LSB Bit in a given state, algorithm toggles LSB Bit in next state.

V. Conclusion

Proposed algorithm needs (n-2) more bits as 'Blk' (blocking) bit for n bit counter. Means total number of required bits are 2(n-1) to implement n bit counter. It takes 2 bit transition lower than normal counter state assignment, independently number of bits of counter. Though better algorithms have been proposed earlier there are more scope to improve this algorithm in terms of reducing 'Blk' bit and bit transition. State assignment algorithms to reduce dynamic power in CMOS circuits can be used in Vehicle electronics to reduce energy consumption in code execution in Processor.

Electric vehicles are future of transportation and number of ECUs is already increasing to make vehicle electronics more power hungry. We can use such algorithms in electric vehicle's ECU so that limited available power supply (Batteries) can be utilize efficiently in transmission, and lowest possible can be used in processing of electric control systems of vehicles.

VI. Acknowledgment

This work was supported by KPIT Technologies Ltd. I am thankful to National Institute of Technology Kurukshetra, Haryana, India for providing me the opportunity to carry on this research work.

References

[1] Duarte L. Oliveira, Tiago Curtinhas, Lester A. Faria, João Luis V. Oliveira, Leonardo Romano, "Design of Low-Power Two-Hot Finite State Machines Operating on Double-Edge Clock," 2016 IEEE.

[2] L. Benini, and G. DeMicheli, "State Assignment for low power Dissipation," IEEE Journ. on Solid State Circ., 11(4): 32-40, March 1994

[3] K. Kajstura, D. Kania, "Binary-Tree base low power state assignment algorithm," International Conference of Computational Methods in Sciences and Engineering 2016 (ICCMSE 2016).

[4] Tiago Curtinhas, Duarte L. Oliveira, Lester A. Faria and Osamu Saotome, "A novel State Assignment method for Extended Burst-Mode FSM design using Genetic Algorithm," 2014.

[5] Chunhong Chen, Jiang Zhao, and Majid Ahmadi, "A Semi-Gray Encoding Algorithm for Low-Power State Assignment," 2003 IEEE.

[6] L. Benini, and G. DeMicheli, "State Assignment for low power Dissipation," IEEE Journ. on Solid State Circ., 11(4): 32-40, March 1994.

[7] D.R. Cox and H.D. Miller, "The Theory of Stochastic Pro-cesses" Chapman Hall, 1965.

[8] DeMicheli, Brayton, Sangiovanni, "Optimal State Ass. for FSMs" IEEE Trans. on CAD, 4(3): 269-284, July 1985.

[9] Najm, Goel, Hajj, "Power Estimation in Sequential Circuits," Proc. of the 32th DAC, 635-640, 1995.

[10] Chen, Sarrafzadeh, Yeap, "State Enc. of FSMs for Low Power Design".

[11] Forth, Molitor, Vogt, "State Encoding of FSMs Target," BDD Repres. Priv. Comm. Univ. Halle, Germany, March 1998.

[12] T. Lengauer: Combinatorial Algorithms for Integrated Cir-cuit Layout. Teubner Verlag, 1990

[13] P. Baccheta, L. Daldoss, D. Sciuto, C. Silvano, "Lower-Power State Assignment Techniques for Finite State Machines," IEEEInternational Symposium on Circuits and Systems, 2000, pp. 641-644.

[14] K. Kajstura, D. Kania, "A decomposition state assignment method of finite state machines oriented towards minimization of Power," Przeglad Elektrotechniczny, no. 6, 2011, pp.146-150.

[15] K. Kajstura, D. Kania, "Low power FSM state assignment using a Kernighan-Lin algorithm," Przeglad Elektrotechniczny, no. 5, 2014, pp.155-159.