Journal of Computer Science 5 (7): 507-510, 2009
ISSN 1549-3636

Using Combinational Circuits for Control Purposes

Maher A. Nabulsi and Mohammed A.F. Al-Husainy
Department of Computer Science, Faculty of Sciences and Information Technology,
Al-Zaytoonah University of Jordan, Amman, Jordan

Abstract: Problem statement: Combinational circuits are used in computers for generating binary control decisions and for providing digital components for data processing. **Approach:** The use of combinational circuits and logic gates to control other circuits was discussed. Different systems that use logic gates, multiplexers, decoders and encoders to control different circuits were presented. This study presented a design and implementation of some combinational circuits such as a decoder, an encoder, a multiplexer, a bus system and read/write memory operations. **Results:** When we connected some types of combinational circuits to the inputs/outputs of digital circuit, these combinational circuits can help us to manage and flow a different types of control signals through a large digital circuit. **Conclusion:** Many combinational circuits had a good function which can be used for controlling different parts of any digital system and they produce a suitable way to transfer a control signals between different digital components of any large digital system.

Key words: Decoder, encoder, multiplexer, register, control unit

INTRODUCTION

Logic circuits for digital systems may be combinational or sequential. A combinational circuit consists of logic gates whose outputs at any time are determined from the present combination of inputs. A combinational circuit performs an operation that can be specified logically by a set of Boolean functions. Sequential circuits such as registers, counters and memories employ storage elements in addition to logic gates. Their outputs are a function of the inputs and the state of the storage elements. The state of storage elements, in turn, is a function of previous inputs. As a consequence, the outputs of a sequential circuit depend not only on present values of inputs, but also on past inputs and the circuit behavior must be specified by a time sequence of inputs and internal states.

A combinational circuit consists of input variables, logic gates and output variables. The logic gates accept signals from the inputs and generate signals to the outputs. This process transforms binary information from the given input data to a required output data. There are several combinational circuits that are employed extensively in the design of digital systems. These circuits are available in integrated circuits and are classified as standard components. They perform specific digital functions commonly needed in the design of digital systems.

Some combinational circuits, such as decoders, encoders and multiplexers, can be used to control other devices such as decoder circuits[1,2], three-state buffers[3,4], register circuits[3], bus circuits, read / write memory operations[5] and others. Decoder and other combinational circuits can be used in mobile system, wireless networks and can be applied for other related communication systems[6,7]. Decoder logic is essential for control units and memories[8].

MATERIALS AND METHODS

Here, we use a simple control unit[3,8-12] to generate control signals, which can be used to control different devices illustrated in our examples. The control unit contains an Instruction Register (IR) to store the instruction code which is read from memory, (4×16) decoder to decode a 4-bit operation code, having only one output (control signal) active at any given time (either D_0 or D_1 or …or D_{15}) and some control logic gates.

We can use a multiplexer to control a register circuit. For example, we use two multiplexers to control a 2-bit register constructed from two D flip-flops as

Corresponding Author: Maher A. Nabulsi, Department of Computer Science, Faculty of Sciences and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan

seen in Fig. 1. This register circuit is controlled by a common selection line of multiplexers (LD):

- If LD = 0 (when D_0 is not active), then A_0 and A_1 are transferred to D flip-flops and we have a no-change state (no loading of new information)
- If LD = 1 (when D_0 is active with operation code 0000), then I_0 and I_1 are transferred to D flip-flops and we have a loading state of new information. This situation is shown in Table 1

A decoder, an encoder and a multiplexer can be used to control some transfers through a bus system. Suppose, for example, it is required to perform the following four transfers through the bus system constructed from multiplexers:

D_1: RB ← RA
D_2: RC ← RB
D_3: RD ← RC
D_4: RA ← RD

Where, RA, RB, RC, RD are 4 bit registers and D_1, D_2, D_3, D_4 are control signals. Figure 2 shows the bus system and the control of the required transfers. In this example, the operation code for the first transfer is (0001), for the second is (0010), for the third is (0011), for the fourth is (0100) and the other operation codes are for other operations in the system. If the operation code is (0001), then D_1 will be active out of the (4×16) decoder, S_0 and S_1 will be equal (00) out of the (4×2) encoder, RA will be transferred to the bus lines, the LD input of RB will be active out of the (2×4) decoder, RB will receive information from the bus lines and the first transfer is performed (D_1: RB ← RA). All other transfers will be performed in the same manner as shown in Table 2.

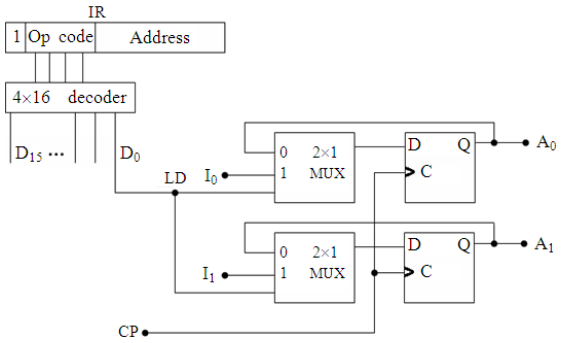

Fig. 1: Two multiplexers with one common selection line to control a 2 bit register

Table 1: Data inputs selected for D flip-flops based on LD control input

Op code	D active	selection input: LD	Output: D	Operation
Any Op code except 0000	$\overline{D_0}$	0	A	Unchanged state
0000	D_0	1	I	loading

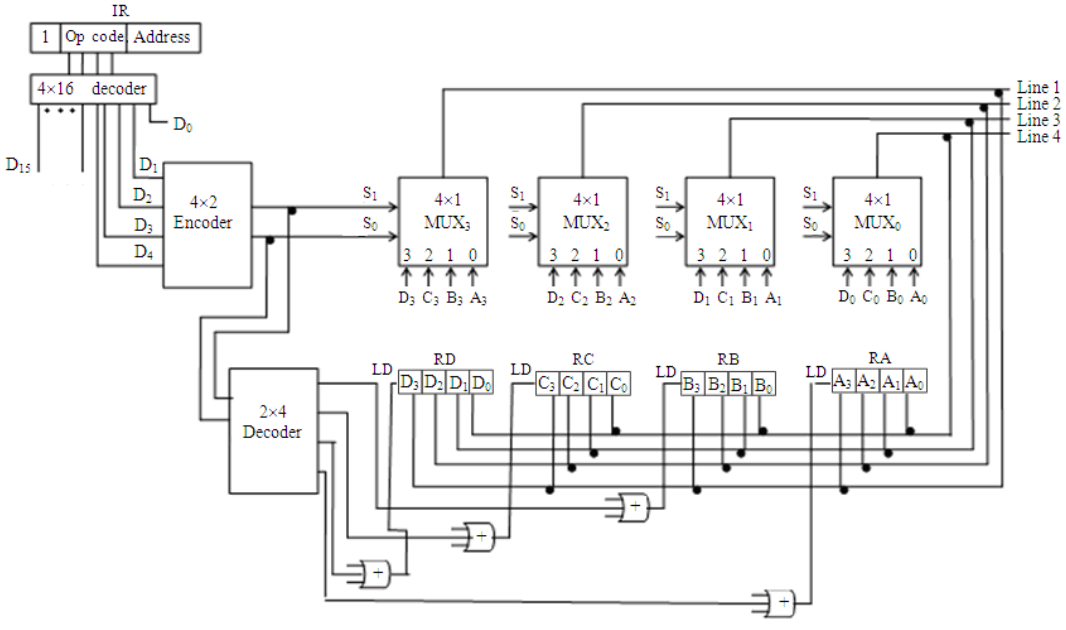

Fig. 2: Using a decoder, an encoder and a multiplexer to control some transfers through a bus system

Table 2: Four transfers through the bus system based on some control inputs

Op. code				D active	S_1	S_0	Selected Reg.	LD active	Operation
0	0	0	1	D_1	0	0	Bus ← RA	LD of RB	RB ← RA
0	0	1	0	D_2	0	1	Bus ← RB	LD of RC	RC ← RB
0	0	1	1	D_3	1	0	Bus ← RC	LD of RD	RD ← RC
0	1	0	0	D_4	1	1	Bus ← RD	LD of RA	RA ← RD

Table 3: Read and write operations on memory

Op. code	D active	Enabled decoder	Operation
0101	D_5	Decoder 1	Write
0110	D_6	Decoder 2	Read

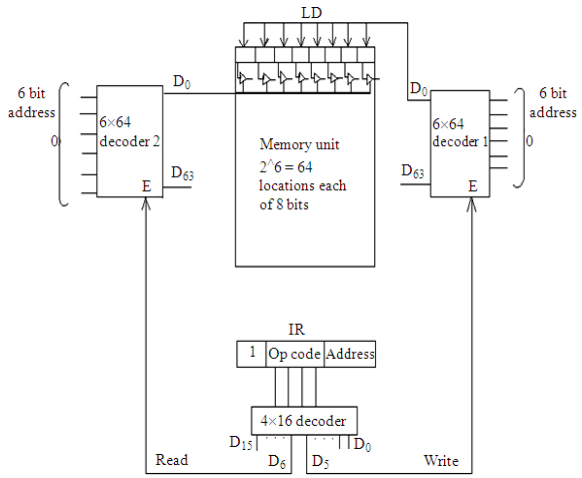

Fig. 3: Two decoders for read and write operations on memory

Data items to be stored (write) in memory or which are retrieved (read) from memory to be used in calculation are sent along the data bus. The address in which that data item is to be stored, or from which it is to be retrieved, is sent along the address bus. All activity within the microcomputer, including the storage and retrieval of data items, is initiated by the microprocessor but managed by the control bus. It is the responsibility of the control bus, for example, to switch the direction of travel on the data bus depending on whether a data item is to be stored or retrieved from memory.

For controlling purposes, we can use two decoders for write to memory and read from memory operations as shown in Fig. 3. In this example, we have a memory unit of 64 locations, each location of 8 bits, constructed from flip-flops with a 6-bit address bus. Here, if the operation code is (0101), then D_5 will be active out of the (4×16) decoder. D_5 is connected to the enable input of Decoder1 for the write to memory operation and D_6 is connected to the enable input of Decoder2 for the read operation through three-state buffers. This situation is shown in Table 3.

Table 4: The control of some inputs of a register (INR and CLR)

Op code	D active	AC test	The active control input of PC
0111	D_7	AC (15) = 0	INR
1000	D_8	AC (15) = 1	INR
1001	D_9	AC = 0	INR
1010	D_{10}	AC = FFFF	CLR

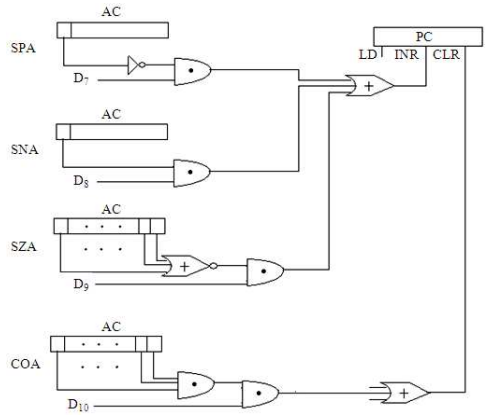

Fig. 4: The implementation of control and execution of some instructions

Also, we can use some logic gates such as AND, NAND, OR, NOR, NOT and others to control and execute some instructions. In this example, we have a 16-bit processor register (accumulator: AC) and a program counter (PC) to hold and calculate the address of instructions. Here, we show how to control the following instructions: SPA (skip next instruction if AC is positive), SNA (skip next instruction if AC is negative), SZA (skip next instruction if AC is zero) and COA (clear PC if AC is all 1's). We specify these instructions with the control signals (D_7, D_8, D_9, D_{10}) and micro-operations as following:

SPA D_7: if (AC (15) = 0), then PC ← PC + 1

SNA D_8: if (AC (15) = 1), then PC ← PC + 1

SZA D_9: if (AC = 0), then PC ← PC + 1

COA D_{10}: if (AC = FFFF), then PC ← 0

The implementation of these instructions is seen in Fig. 4. The conditions necessary to set the required control input of PC are shown in Table 4.

RESULTS

Some combinational circuits such as a decoder, an encoder and a multiplexer were used to control other devices:

- Using multiplexers to control the load operation of a register
- Using a decoder, an encoder and a multiplexer to control some transfers through the bus system
- Using two decoders to control the read and write operations on memory
- Using some logic gates to control the INR and CLR inputs of a register

DISCUSSION

This study tries to focus on the usefulness of some common combinational circuits (i.e., a decoder, an encoder and a multiplexer) in controlling many different components and operations (register, memory, bus, and execution of instructions) of a digital computer. Through Fig.1 to Fig. 4, a novel suggestion of hardware connections between a set of decoders, encoders and multiplexers with different components in a digital computer is presented. These connections helped the system generate many control signals to perform different operations. This demonstration sheds light on the importance of these combinational circuits in control. We recommend employing these combinational circuits in controlling other components of computer systems (like the Arithmetic Logic Unit (ALU), input/output units, etc.) and communication systems.

CONCLUSION

In this study, a small architecture of decoder, encoder and multiplexer for register, bus system and memory operations was presented. We conclude that the use of these combinational circuits can help the designer of the digital computers to generate different types of digital signals for controlling many components in the digital computer.

REFERENCES

1. Stephen Brown and Zvonko Vranesic, 2005. Fundamentals of Digital Logic with VHDL Design. 2nd Edn., McGraw-Hill Higher Education, USA., ISBN: 0072499389, pp: 939.
2. Wakerly, J.F., 2006. Digital Design-Principles and Practices. 4th Edn., Pearson Prentice Hall, USA., ISBN: 0132128381, pp: 859.
3. Mano, M.M., 1993. Computer System Architecture. 3rd Edn., Prentice Hall, USA., ISBN: 0-13-175738-5, pp: 525.
4. Nabulsi, M.A. and A.M. Abdalla, 2008. The relationship between exclusive-or and the unique existential quantifier. J. Comput. Sci., 4: 741-743.
5. Mano, M.M., 2002. Digital Design. 3rd Edn., Prentice Hall, USA., ISBN: 0-13-062121-8.
6. Chaiwat Keawsai, Keattisak Sripimanwat and Attasit Lasakul, 2004. Modified register exchange method of viterbi decoder for 3GPP mobile system. ECTI, Pataya, Thailand. http://www.kmitl.ac.th/dslabs/download/paper/viterbidecoder_ecti2004.pdf
7. Zhang, L., Y.J. Cheng and X. Zhou, 2009. Rate avalanche: Effects on the performance of multi-rate 802.11 wireless networks. Simulat. Model. Pract. Theor., 17: 487-503. DOI: 10.1016/j.simpat.2008.09.003
8. Rob Williams, 2001. Computer Systems Architecture. Addison-Wesley, ISBN: 0201648598.
9. Stallings, W., 2006. Computer Organization and Architecture Designing for Performance. 7th Ed. Pearson Prentice Hall, USA., ISBN: 0131856448, pp: 778
10. Giovanni De Micheli, 1988. Computer-aided design and optimization of control units for VLSI processors. Int. J. Circ. Theor. Appli., 16: 347-369.
11. Tang, K.S., K.F. Man, G. Chen and S. Kwong, 2001. An optimal fuzzy PID controller. IEEE. Trans. Ind. Elect., 48: 757-765. DOI: 10.1109/41.937407
12. Zhixiong, Y., Z. Xuecheng, L. Weizhong and C. Weibing, 2006. A multiplexing algorithm of multiple elementary streams based on virtual buffer control. Wuhan Univ. J. Nat. Sci., 11: 625-630. DOI: 10.1007/BF02836678