Design of Control Architecture for Industrial Process Automation

Ihedioha A.C, Eneh I. I, and Onoh G.N.,
Department of Electrical/Electronics Engineering, Enugu State University of Science and Technology, Enugu, Nigeria.

Abstract: This paper is on the design of control architecture for industrial process automation. An analysis was done using a typical process control design example involving palm fruit loading. The development of a structured software engineering approach to process control based on the state transition table and the compact Read Only Memory (ROM) based architecture that features a content addressable memory and input/output line reduction technique was achieved. The research therefore recommend to researchers, development institutes and industrial equipment developers/builders, especially those committed to the realization of vision 20: 2020 in Nigeria to make use of the industrial process automation developed in this research.

Keywords*: Automation, Industrial, Read Only Memory (ROM), Architecture, Nigeria*

I. INTRODUCTION

Vision 20:20 in Nigeria is an embodiment of the nation’s objective to become one of the top industrialized nations of the world by the year 2020. The launching of this vision attests to the fact that Nigeria is not happy to find herself merely as a consumer nation with respect to industrial products, despite being a giant in Africa.

Also, there has been a strong impetus to look inwards towards indigenous efforts at high technology acquisition rather than continue to hope that technology shall be transferred to her willingly by industrialized nations [1].

Therefore, a push within the nation for an industrialization process that would be spear headed by and sustained with locally available expertise and with the help of Nigerians in Diaspora became necessary. A number of research efforts are therefore ongoing in response to this national call and an example is the design and manufacture of 80C engine motorcycle at the National Engineering Design Development Institute (NEDDI), Nnewi, Nigeria, by Nigerian engineers using local materials, as one of the targets of vision 20:2020 economic reformation agenda.

This research is occasioned by the need to develop new and improved architecture for digital process control and monitoring systems, a case study of a composite palm fruit processing machine.

II. A TYPICAL PROCESS CONTROL

A common example of process control is seen in the palm fruits processing machine. Several efforts had been made in the past to automate these processes [2]. But a lot of refinements are still needed giving more room for further research.

Figure 1 shows a simplified diagram of a composite palm fruit loading, digestion and filtering machine. The HREADY position of the machine (i.e. when it is ready for the next loading, digestion and filtering process) is when digester door is closed and separator door is closed and there is no palm fruit in the digester.

This machine is implemented using various types of logic devices in order to showcase their architecture. The type of logic element used in the implementation of this palm fruit processing system determines the complexity of the resultant system measured in terms of the number of logic elements used (i.e. chip count), the number of signal lines interconnected in the implementation, its maintainability, scalability, flexibility and potential sources of error[4].

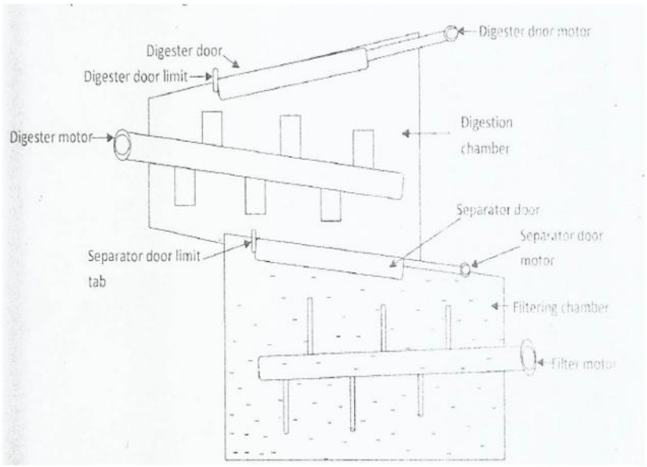
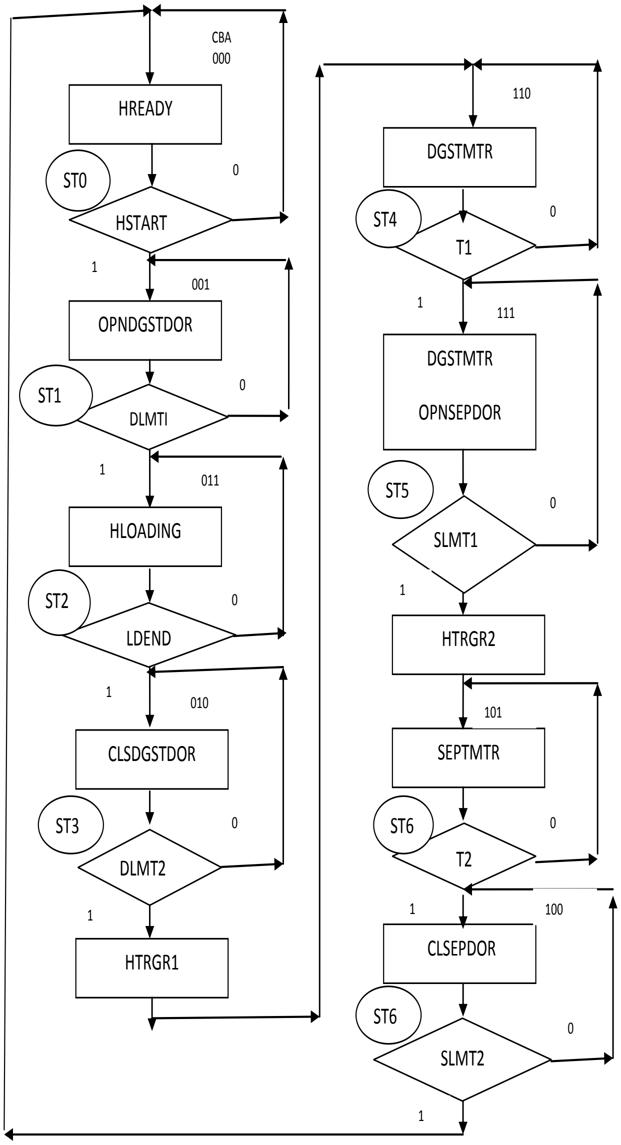


Figure 1: A Composite Fruit Processing Machine [3].

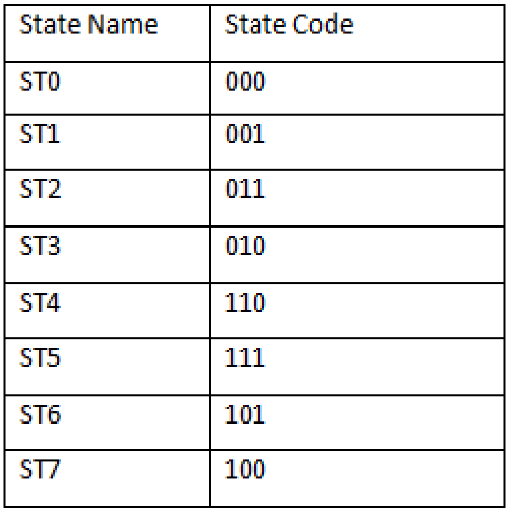
III. IMPLEMENTATION

The ASM chart for loading, digesting and filtering with the state map and State Assignment table is shown in figure 2. Table 1 is the state transition table corresponding to the Algorithmic State Machine (ASM) chart of figure 2. Clearly, the qualifier field is 8bits long to accommodate the signals: HSTART, DMLT1, LDEND, DLMT2, T1, SLMT1, T2, SLMT2. The state code is 3 bits long to accommodate 8 unique binary combinations, one for each of the states ST0 through ST7. If a microprocessor based control system or ROM based control system is desired, the address inputs to the memory containing the process information would be 11 bits long because the qualifier fields (8 bits) is concentrated with present state code (3 bits) to form the address.



CB \ A	0	1
00	ST0	ST1
01	ST3	ST2
11	ST4	ST5
10	ST7	ST6

Figure (a): State Map



State Name	State Code
ST0	000
ST1	001
ST2	011
ST3	010
ST4	110
ST5	111
ST6	101
ST7	100

Figure (b): State Assignment

Figure 2: ASM Chart for Loading, Digesting and Filtering With the State Map and State Assignment Table

The process can be explained as follows:

1. At state ST0, the HREADY signal is on and the control system is waiting for a start signal (HSTART) from the operator. For as long as the operator has not pressed HSTART, the HREADY signal is on. As soon as HSTART button is pressed, the control system transits to ST1.

2. At ST1, the signal to open digester door (OPNDGSTDOR) is on and this initiates the door opening process. The control system begins to sense the door limit switch (DLMT1) which is logic zero when the door is not fully open and logic one when the door is fully open. When the DLMT1 signal is received as logic 1 the control system moves to state ST2.

3. At state ST2, the control system waits for the palm fruit to be loaded into the upper compartment of the composite machine which is called the digestion chamber. The operator loads the palm fruit into the digestion chamber and presses a button (LDEND) to signify end of loading. For as long as that button is not pressed (i.e. LDEND=0) the control system remains in state ST2 but as soon as it is pressed (i.e. LDEND=1) the control system transits to state ST3.

4. At ST3, the control system initiates the closing of the digester door using the signal CLSDGSTDOR. When the digester door is fully closed it would make contact with a door limit switch (DLMT2). If contact is not yet made, DLMT2=0 and when contact is made DLMT2=1. The signal to close digester door remains on until DLMT2=1 whereupon the control system transits to ST4. En-route to ST4, it outputs a conditional output HTRGR1 to signal the end of loading and to trigger the start of time interval T1 that controls the duration of palm fruit digestion.

5. At ST4, the digester motor DGSTMTR is turned on to start the automated pounding of the palm fruit. Through experimentation, it is found to require T1 seconds to be completed. For as long as time spent in digesting is less than t1, the T1 signal is zero. Immediately the time is up to T1, the control system transits to state ST5.

6. At ST5, the digester motor is left on still, while the control system initiates the opening of the door between the digester chamber and filtering chamber called OPNSEPDOR. This is to allow the digested palm fruit to fall through into the filtering chamber which is typically immersed in water. The separator door limit switch SLMT1 remains zero for as long as the separator door is not fully open. It jumps to logic 1 immediately the separator door is fully open. This causes the control system to output a conditional output signal HTRGR2 to signal the end of digestion and to initiate the timing intervalT2 needed for filtering. The control system then transits to ST6.

7. At ST6, the filter motor is turned on using the signal FLTRMTR. The filtering process automates the separation of the fibre from the kernel with finger-like protrusions from the spindle. This is done under water and in the process the oil washes off into the water which is removed later for further processing. The filtering process has been predetermined to last for T2 seconds through experimentation. While the time elapsed during filtering is less than T2, T2=0. Immediately the time of filtering is equal to T2 seconds, the control system transits to St7.

8. At ST7 the processor initiates the closing of the separator door (CLSEPDOR) watching out for door limit switch signal (SLMT2) which indicates when the separator door is not fully closed. For as long as the separator door is not fully closed, SLMT2=0. Immediately the separator door is fully closed SLMT2=1 whereupon the control system transits to state ST0 to indicate READY for the next production run.

Table 1: State Transition Table for Palm Fruit Process Control

Link Path	**Qualifiers**								**Presen t State Code**	**Next State Code**	**State Outputs**								**Condi-tional output**	
	HSTART	DLMT1	LDEND	DLMT2	T1	SLMT1	T2	SLMT2	CBA	CBA	HREADY	OPNDGSTDR	HLOADG	CLSDGSTDR	DGSTMTR	OPNSEPDRR	FLTRMTR	CLSEPDR	HTRGR1	HTRGR2
L1	0	-	-	-	-	-	-	-	0 0 0	0 0 0	1	0	0	0	0	0	0	0	0	0
L2	1	-	-	-	-	-	-	-	0 0 0	0 0 1	1	0	0	0	0	0	0	0	0	0
L3	-	0	-	-	-	-	-	-	0 0 1	0 0 1	0	1	0	0	0	0	0	0	0	0
L4	-	1	-	-	-	-	-	-	0 0 1	0 1 1	0	1	0	0	0	0	0	0	0	0
L5	-	-	0	-	-	-	-	-	0 1 1	0 1 1	0	0	1	0	0	0	0	0	0	0
L6	-	-	1	-	-	-	-	-	0 1 1	0 1 0	0	0	1	0	0	0	0	0	0	0
L7	-	-	-	0	-	-	-	-	0 1 0	0 1 0	0	0	0	1	0	0	0	0	0	0
L8	-	-	-	1	-	-	-	-	0 1 0	1 1 0	0	0	0	1	0	0	0	0	1	0
L9	-	-	-	-	0	-	-	-	1 1 0	1 1 0	0	0	0	0	1	0	0	0	0	0
L10	-	-	-	-	1	-	-	-	1 1 0	1 1 1	0	0	0	0	1	0	0	0	0	0
L11	-	-	-	-	-	0	-	-	1 1 1	1 1 1	0	0	0	0	0	1	0	0	0	0
L12	-	-	-	-	-	1	-	-	1 1 1	1 0 1	0	0	0	0	0	1	0	0	0	1
L13	-	-	-	-	-	-	0	-	1 0 1	1 0 1	0	0	0	0	0	0	1	0	0	0
L14	-	-	-	-	-	-	1	-	1 0 1	1 0 0	0	0	0	0	0	0	1	0	0	0
L15	-	-	-	-	-	-	-	0	1 0 0	1 0 0	0	0	0	0	0	0	0	1	0	0
L16	-	-	-	-	-	-	-	1	1 0 0	0 0 0	0	0	0	0	0	0	0	1	0	0

IV. FACTORS THAT INFLUENCE CHOICE OF LOGIC DEVICE

The choice of logic systems to use in a given application is a function of the following factors:

1. The level of complexity of the logic system. The more complex a system is, the more one moves away from random logic into more structured hardwired logic or into software based systems if the system is very complex [5].

2. After sales maintainability issues and system reliability. This influence the choice of logic system to use because the better the ease of maintenance the greater the choice of the logic system.

3. Cost of the logic system. Any logic designer would try to avoid an unnecessary 'overkill' in price as this would eventually affect the product price which would be a disadvantage should the product compete unfavourably in the market. Hence an equivalent logic system which satisfactorily serve the same purpose and less costly should be used.

4. Component device availability issues. When procurement of a device or component becomes an uphill task, it can hamper on the time constraint of the job at hand or totally stop the progress of the work [6]. Hence a designer can adapt what is readily available rather than hoping for the best alternative that may never be sourced.

CONCLUSION

In this paper, the design of control architecture for industrial process automation has been presented. The researcher developed an architecture for industrial process automation in which digital means are used for process control and monitoring. Palm fruit process control and monitoring was used as a case study. The compact ROM-based architecture that features a content addressable memory and input/output line reduction technique was achieved. It therefore recommend to researchers, development institutes and industrial equipment developers/builders, especially those committed to the realization of vision 20: 2020 in Nigeria to make use of the industrial process automation developed in this research.

References

[1] Moreau L 2002 Agents for the grid: a comparison for Web services (Part 1: the transport layer). In Second IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002) (eds. Bal HE, Lohr KP and Reinefeld A), pp. 220-228. IEEE Computer Society, Berlin, Germany.

[2] Moreau L, Miles S, Goble C, Greenwood M, Dialani V, Addis M, Alpdemir N, Cawley R, Roure DD, Ferris J, Gaizauskas R, Glover K, Greenhalgh C, Li P, Liu X, Lord P, Luck M, Marvin D, Oinn T, Paton N, Pettifer S, Radenkovic MV, Roberts A, Robinson A, Rodden T, Senger M, Sharman N, Stevens R, Warboys B, Watson P and Wroe C 2002 On the Use of Agents in a BioInformatics Grid. Network Tools and Applications in Biology (NETTAB'2002) – Agents in Bioinformatics, Bologna, Italy.

[3] Muscettola N, Nayak PP, Pell B and Williams B 1998 Remote agent: To boldly go where no AI system has gone before. Artificial Intelligence**103**(1-2), 5-48.

[4] Nowostawski M, Purvis M and Cranefield S 2001 A layered approach for modelling agent conversations, Proceedings of the 2nd International Workshop on Infrastructure for Agents, MAS, and Scalable MAS, 5th International Conference on Autonomous Agents, pp. 163-170. Montreal, Canada.

[5] Odell J, Parunak H and Bauer B 2000 Extending UML for agents. Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence, pp 3-17.