DESIGNING MULTI-AGENT BASED LINKED STATE MACHINE

Hyacinth C. Inyiama[1], Ifeyinwa Obiora-Dimson[2], Christiana C. Okezie[3]

[1, 2, 3] Department of Electronic and Computer Engineering Nnamdi Azikiwe University, Awka
Anambra State, Nigeria. Phone: +2348032638315
hcinyiama@gmail.com, ifeyinwa29@yahoo.com, christianaobioma@yahoo.com

Abstract

Every industrial process control algorithm can be represented as an ASM chart and every ASM chart can be implemented using the state agent based approach. This paper describes a simplified approach to implementing complex process control systems using the agent-based paradigm. It shows that when the system is complex, it can be broken up into a series of ASM charts, and each of those ASM charts can be implemented using agent-based approach comprising of one process agent and a number of state agents. These agent based systems together perform the complex industrial process control. This structure can be interconnected to form a linked state machine in order to realize a multi-agent based linked state structure. This is a 'divide and conquer' approach that is much simpler than tackling the control system implementation as one very complex effort.

Index Terms: *Process control, ASM chart, State Transition Table (STT), Multi-agent, Link state machines, Beverage blending*

---***---

1. INTRODUCTION

There are several ways of tackling the complexity associated with complex process control systems. One of such methods is through the use of linked state machines as showcased by Akpado et al (2010) and also by Inyiama et al (2011). The former suggested a way to handle the automation of three individual state machines of an industrial process control separately and then linked them together to function as a complex system. The later in their approach showed how an ASM chart of a complex control system can be simplified by breaking up the entire control logic into smaller subsystems or state machines. The individual state machines were implemented using a ROM and then linked up together to perform the process control. The use of ASM chart and the corresponding State Transition Table (STT) approach enforced a disciplined approach to the realization of the system.

Inyiama et al (2012) also described the agent based architecture for process control system. The research showed how each state in an ASM chart is assigned to an agent. Hence for an ASM chart with n number of states there also exists n numbers of state agents. A process agent was also defined to handle the complexities that may arise in carrying out the process control by the multi-agents. The agent classes where defined to comprise of classes 0, 1, 2, 3 based on the number of input qualifiers between the states. Furthermore the agent software was defined and the complex system was made to fit properly into a standard microcontroller. Where the input/output line is more than 1 byte i.e. 8, line reduction is achieved through input multiplexing and output decoding as demonstrated by Inyiama et al (in print)

This paper extends the technique of linked state machines; by making each machine in the link an agent based system. I/O line reduction technique was used where necessary to fit the system into a microcontroller for individual agent based state machines. Then the agent based state machines are linked up as one complex process controller with easily implemented components.

2. INDUSTRIAL PROCESS EXAMPLE

A certain industrial beverage blender shall be used to showcase the linked state agent based control system design. The structure of the beverage blender comprises of three main parts namely the upper tanks, the lower tanks and the mixers. The upper tanks (fig 1) are where the raw beverages needed for the blending process are poured into otherwise referred to as the reservoir.

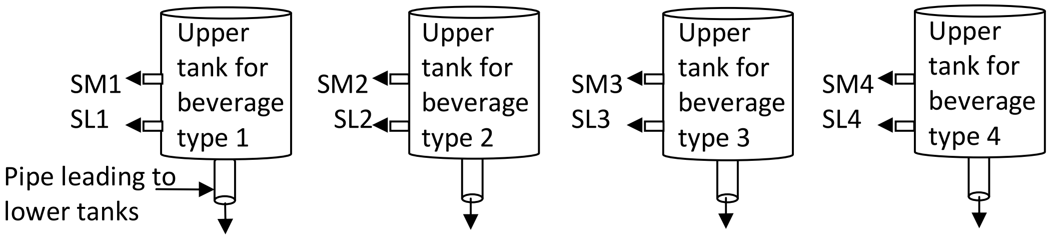

Upper tank for beverage type 1
Upper tank for beverage type 2
Upper tank for beverage type 3
Upper tank for beverage type 4
SM1
SL1
SM2
SL2
SM3
SL3
SM4
SL4
Pipe leading to lower tanks

Fig-1: Upper tanks of an industrial beverage blender

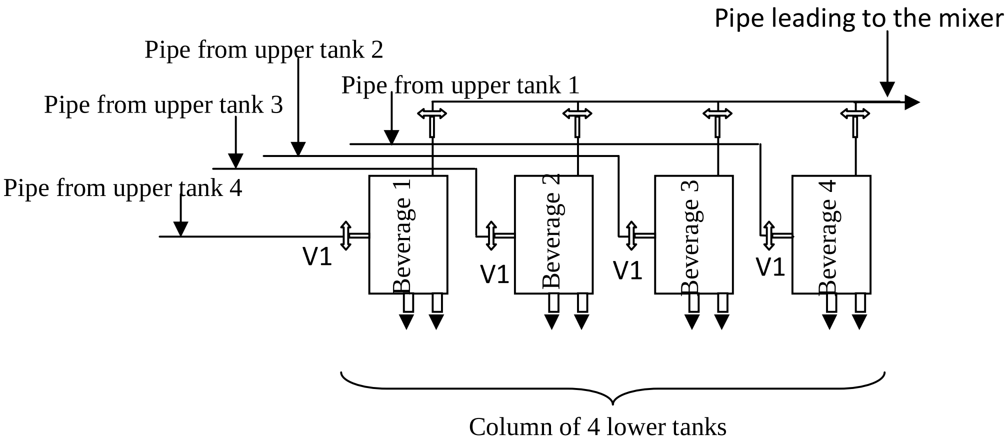


Fig-2: Column of lower tanks of an industrial beverage blender

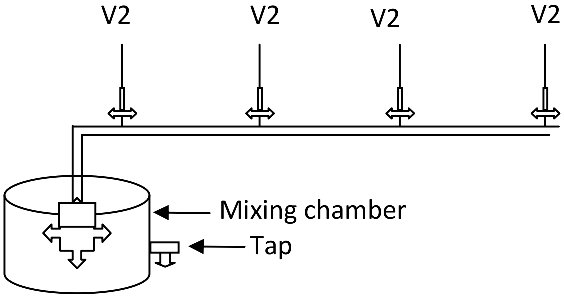


Fig 3: The mixer of the beverage blender

Here four different raw beverage types is required hence the use of four upper tanks. Each of these tanks has an outlet which allows flow to a group of 4 lower tanks arranged in columns as shown in figure 2. Each lower tank has 2 valves which control inlet from the upper tank and outlet to the mixer (fig 3). Hence the required ratio of beverage type 1, 2, 3, and 4 are let down through their respective valves automatically from each of the tanks to the mixer. Then this quantity of beverage is blended at the mixing chamber and then extruded through the tap as shown in fig 3.

A composite structure of this industrial blender is as shown in fig 4. The four boxes above in the diagram represent four upper tanks whose functions have been described in fig 1. Each column of four boxes represents the lower beverage tanks used for the blending processes. They serve the purpose of taking a known quantity of each beverage type before they are used in blending processes. The four boxes in a column are fed from the four upper tanks respectively. Thus they are used to keep track of the quantity of each beverage type used at that column's blending station. The four columns of lower tanks are identical. In each column, the first tank is connected to upper tank 1, the second tank to upper tank 2, the third tank to upper tank 3, the fourth tank to upper tank 4.

Each of the lower tanks has two valves labeled V1 and V2 respectively. V1 in each case controls the refill of the lower tank from the corresponding upper tank, while V2 controls the quantity of each lower tank that is involved in blending at the mixer. Thus, the main difference between one column of beverage box and another is the blending ratios. The blending ratio used for column 1 differ from that of column 2 and so on. Therefore mixers 1, 2, 3, 4 produce different blends of the beverage in upper tanks 1, 2, 3, 4. The number of times each of the four lower tanks in a column is refilled is tracked so as to quantify the volume of each beverage type used at mixer 1. Each of the lower tanks has a fixed volume, 10 litres say. Therefore by multiplying the number of lower tank refills for each of the beverage types with the volume for that, the volume of each beverage type used at mixer 1 is determined. The same is true of volume of lower tank leading to 2, 3, 4) respectively. The difference in the outputs of mixers 1 through 4 is dependent on the blending ratio used for each column of lower tanks.

Note that the output from each mixer stations can also be determined using the number of runs and blending ratios for each mixer station. Thus, the system is able to track both the volume of input and volume of output. This makes it possible for management to determine any pilfering of input (raw material) or output (finished product) done by the work force during the shift.

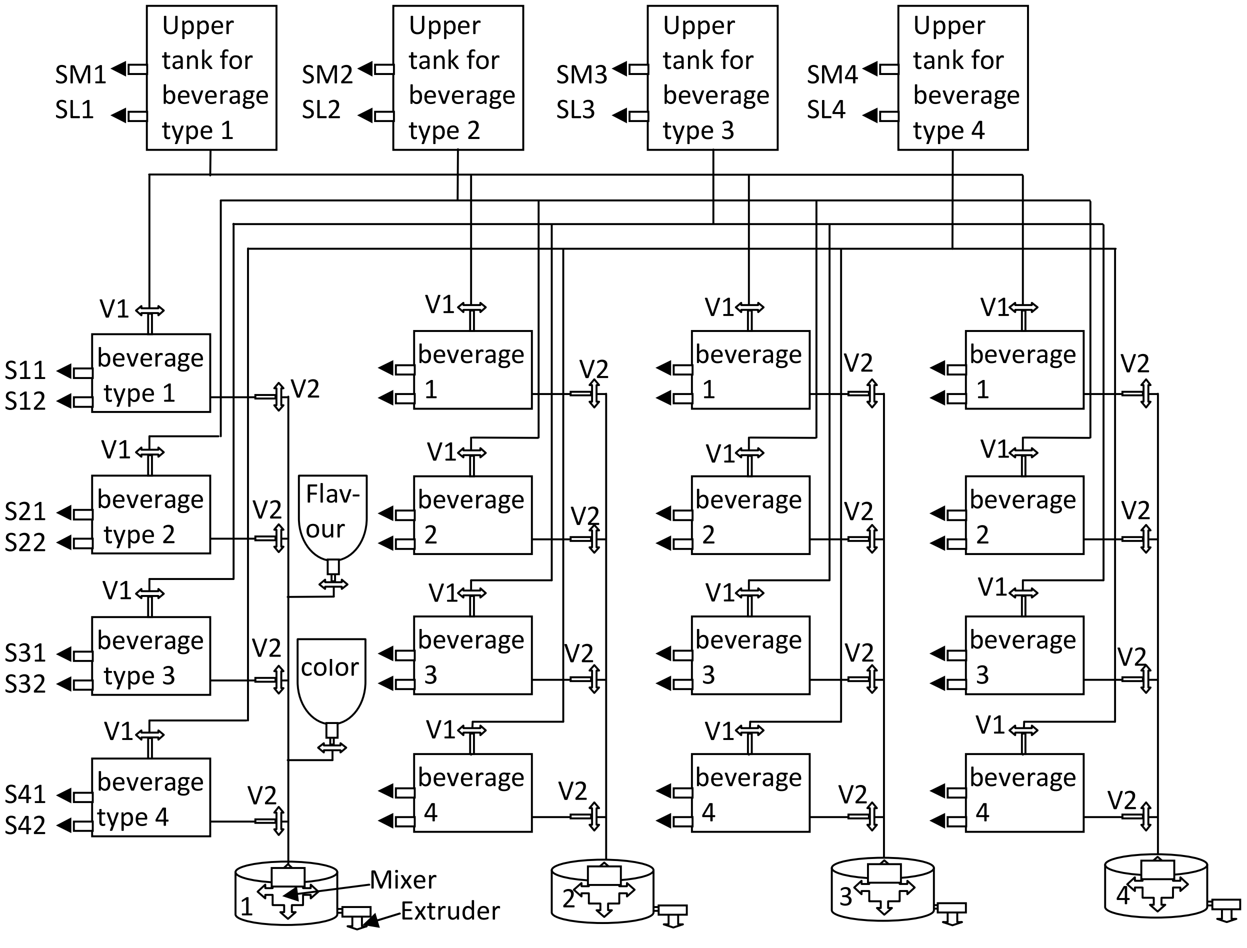


Fig 4: Automated blender/mixer for beverages and fruit juices

3. SYSTEM DESIGN

Realization of the beverage blending operation described in figure 4 involves three processes namely:

a. Upper tank control: co-ordinates the filling of the upper tank to the adequate level and indicating to the operators when it is low and requires refilling.
b. The beverage level sensing and refill control: ensures proper functioning of the lower tanks, the valves controlling inflow and outflow, and proper

dispensing of desired volumes of beverages into the mixing chamber.

c. The beverage blending operation: cause appropriate ratios of pure beverages to flow from the lower tanks into its mixing chambers, blends and extrudes same.

These three processes each can be handled as one agent based state machine which would then be linked up to function as a composite machine using the following agent based linked state design steps:

- Develop an ASM chart from a given industrial process problem
- Break down the ASM into simpler state machines to manage the complexity
- Develop the STT for each state machine
- Modify their state transition table to accommodate input multiplexing where appropriate and to produce a fully expanded STT that can be indexed with the state code
- Assign a process agent for each state machine and a state agent for each of its states.
- Define agent software
- Fit the design of each state machine into a microprocessor or microcontroller
- Apply further output line reduction where necessary using output decoding if necessary
- Then link the state machines together as multi-agent based linked state machines.

4. UPPER TANK CONTROL

The control algorithm for the 4 upper beverage tanks is shown in the ASM chart of figure 5 together with the state map and state assignment. It is designed that a signal STOPRUN be sent to the beverage blending positions whenever any of the four upper tanks is low on stock. A STOPALARM is also issued when this happens. The workforce responsible for feeding the upper tanks with pure beverages is alerted when any of the tanks is midway full to enable them process more fruits for that tank before it runs out of stock.

The description of the ASM is as follows:

- At state ST0, the control system checks if beverage tank 1 is low (BVTNK1L). If so, it triggers a STOPALARM and generates a STOPRUN signal at state ST1. It remains in that loop containing ST1 until the situation is remedied.

Again if at ST0, the beverage tank 1 is not low, the system checks if the stock has run down to half tank (BVTNK1M). If BVTNK1M is not yet at the middle position the control system transits to state ST2. If however the beverage tank 1 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know that the tank would run out of stock if more fruit juice is not put there soon enough. The control system then transits to state ST2.

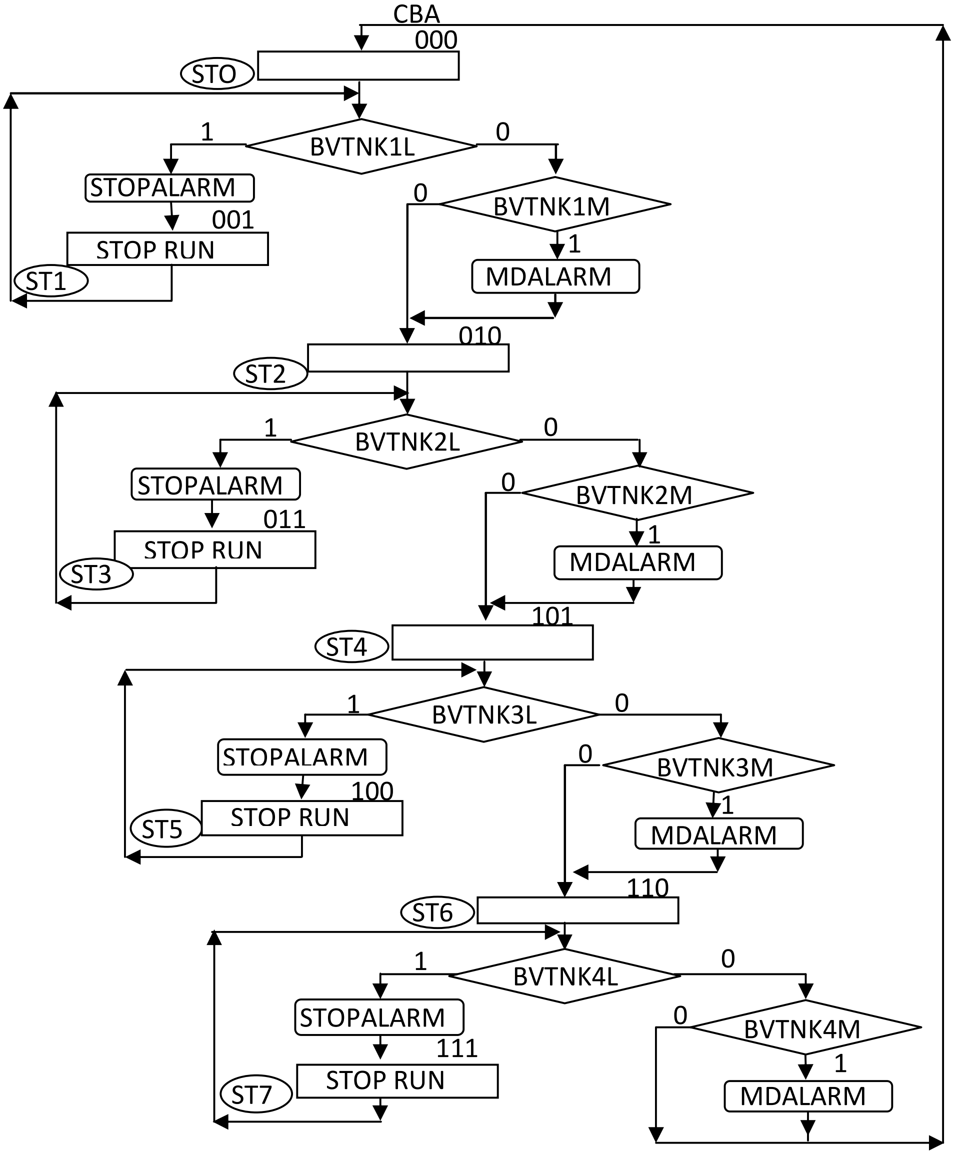
- At state ST2, the control system checks if beverage tank 2 is low on stock (BVTNK2L) if so, it triggers a STOPALRM and generates a STOPRUN at state ST3. It remains in that loop containing ST3 until the situation is remedied.

Again if at ST2 the beverage tank 2 is not low, the system checks if the stock has rundown to half/middle tank (BVTNK2M). If BVTNK2M is not yet at the middle position, the control system transits to state ST3. If however the beverage tank 2 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know that the tank would run out stock. The control system then transits to the state ST4.

- At ST4 the control system checks if beverage tank 3 is low on stock (BVTNK3L). If so, it triggers a STOPALARM and generates a STOPRUN at state ST5. It remains in that loop containing ST5 until that situation is remedied.

Again, if at ST4, the beverage tank 3 is not low, the system checks if the stock has rundown to middle of tank (BVTNK3M). If BVTNK3M is not yet at the middle position, the control system transits to state ST6. If however the beverage tank 3 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so they know the tank would soon run out of stock. The control system then transits to state ST6.

- At state ST6, the control system checks if beverage tank 4 is low on stock (BVTNK4L) if so, it triggers a STOPALARM and generates a STOPRUN at state ST7. It remains in that loop containing ST7 until the situation is remedied. If at ST6, the beverage tank 4 is not low, the system checks if the stock has rundown to middle of tank (BVTNK4M). If BVTNK4M is not yet at the middle position, the control system transits to state ST0. If however the beverage tank 4 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know the tank would run out of stock soon. The control system then transits to state ST0 and the entire cycle is repeated.



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

State Map

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

State assignment

Fig-5: ASM Chart for upper tank control

The state transition table corresponding to the ASM chart of figure 5 is shown in table 1. Here there are 11 input lines comprised of eight qualifiers (BVTNK1L, BVTNK1M, BVTNK2L, BVTNK2M, BVTNK3L, BVTNK3M, BVTNK4L, BVTNK4M), and three present state codes bits (C, B, A). For a fully expanded state transition table, there would be 211=2048 address lines to decode. The diagram of fig 6 shows how input multiplexing can be used to reduce the direct qualifier input to two. In this implementation, the four low level sensor signals are multiplexed such that one is selected at a time. This permits the presentation to the control system of one low level sensor signal and corresponding midpoint sensor signal at one and the same time is required by many of the link paths of figure 5 (the ASM chart). Another advantage of fig 6 is that the number of address lines is now 5 instead of 11 and the control pattern will now comprise 25=32 rows. Thus, the clever application of multiplexing has significantly reduced the ROM memory demand of the application.

Table-1: STT for upper tank control

Link Path	Qualifiers								Present state		Next state		State output	Conditional output	
	BVTNK1L	BVTNK1M	BVTNK2L	BVTNK2M	BVTNK3L	BVTNK3M	BVTNK4L	BVTNK4M	Name	Code	Name	Code	STOPRUN	STOPALARM	MDALARM
L1	0	0	-	-	-	-	-	-	ST0	000	ST2	010	0	0	0
L2	0	1	-	-	-	-	-	-	ST0	000	ST2	010	0	0	1
L3	1	-	-	-	-	-	-	-	ST0	000	ST1	001	0	1	0
L4	0	0	-	-	-	-	-	-	ST1	001	ST2	010	1	0	0
L5	0	1	-	-	-	-	-	-	ST1	001	ST2	010	1	0	1
L6	1	-	-	-	-	-	-	-	ST1	001	ST1	001	1	1	0
L7	-	-	0	0	-	-	-	-	ST2	010	ST4	100	0	0	0
L8	-	-	0	1	-	-	-	-	ST2	010	ST4	100	0	0	1
L9	-	-	1	-	-	-	-	-	ST2	010	ST3	011	0	1	0
L10	-	-	0	0	-	-	-	-	ST3	011	ST4	100	1	0	0
L11	-	-	0	1	-	-	-	-	ST3	011	ST4	100	1	0	1
L12	-	-	1	-	-	-	-	-	ST3	011	ST3	011	1	1	0
L13	-	-	-	-	0	0	-	-	ST4	100	ST6	110	0	0	0
L14	-	-	-	-	0	1	-	-	ST4	100	ST6	110	0	0	1
L15	-	-	-	-	1	-	-	-	ST4	100	ST5	101	0	1	0
L16	-	-	-	-	0	0	-	-	ST5	101	ST6	110	1	0	0
L17	-	-	-	-	0	1	-	-	ST5	101	ST6	110	1	0	1
L18	-	-	-	-	1	-	-	-	ST5	101	ST5	101	1	1	0
L19	-	-	-	-	-	-	0	0	ST6	110	ST0	000	0	0	0
L20	-	-	-	-	-	-	0	1	ST6	110	ST0	000	0	0	1
L21	-	-	-	-	-	-	1	-	ST6	110	ST7	111	0	1	0
L22	-	-	-	-	-	-	0	0	ST7	111	ST0	000	1	0	0
L23	-	-	-	-	-	-	0	1	ST7	111	ST0	000	1	0	1
L24	-	-	-	-	-	-	1	-	ST7	111	ST7	111	1	1	0

The fully expanded state transition table corresponding to figure 6 is shown in table 2. The foregoing shows that input multiplexing and output decoding can be applied to reduce the complexity of any sequential control system but the exact approach would differ depending upon the exact demand of each application. Two of such approaches have been demonstrated namely: Using only one multiplexer at the input if only one qualifier is required at a time. (fig 8) And using more than one multiplexer (fig 6).

This makes the approach quite universal as it can be adapted to match the needs of any particular control application.

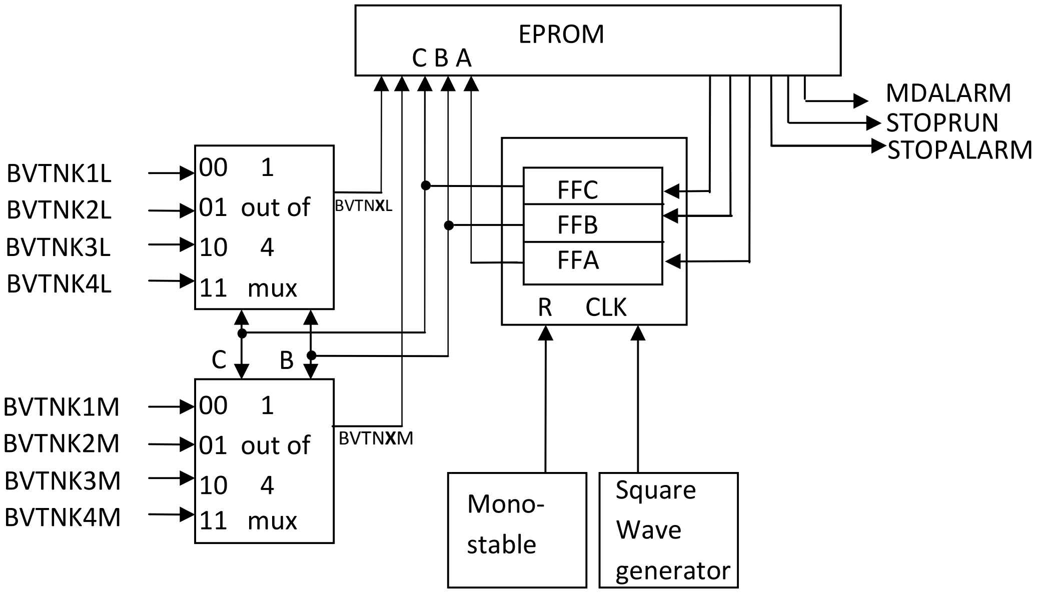


Fig-6: ROM based upper tank control using more than one input multiplexer

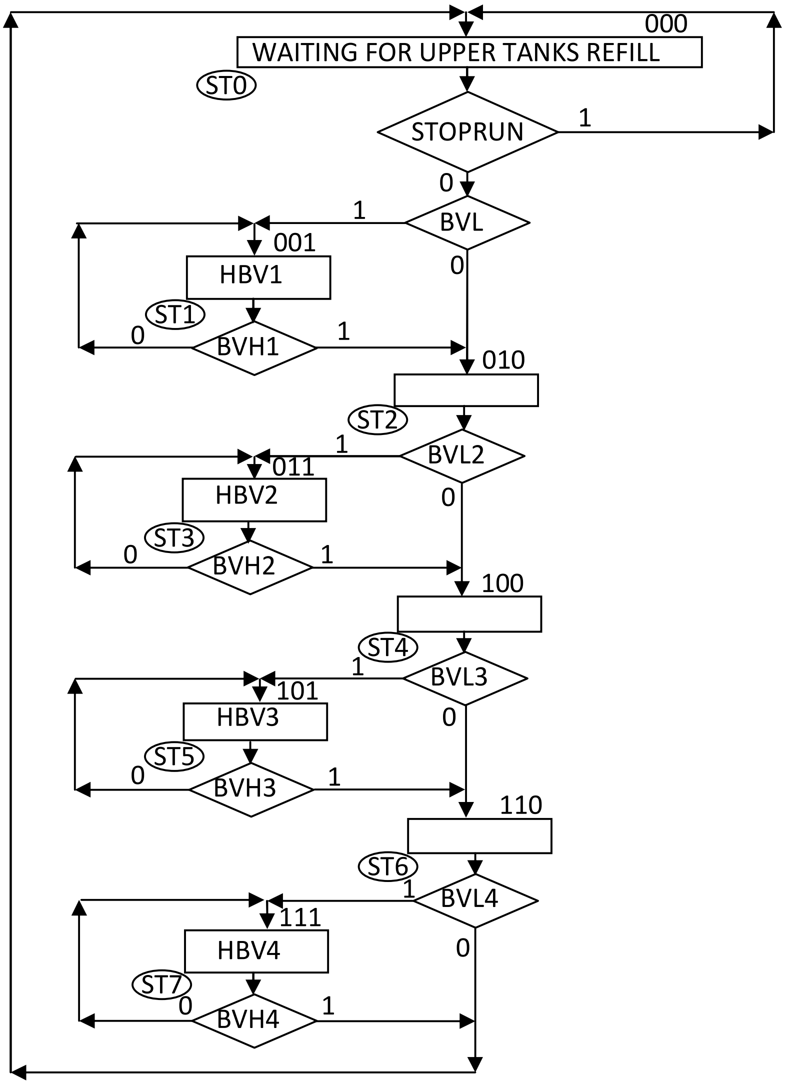
Table 2: STT for the optimized ROM solution

Link Path	Qualifiers		Present state		Next state		State output	Conditional output	
	BVTNKXL	BVTNKXM	Name	Code	Name	Code	STOPRUN	STOPALAR M	MDALARM
L1	0	0	ST0	000	ST2	010	0	0	0
L2	0	1	ST0	000	ST2	010	0	0	1
L3	1	0	ST0	000	ST1	001	0	1	0
L3	1	1	ST0	000	ST1	001	0	1	0
L4	0	0	ST1	001	ST2	010	1	0	0
L5	0	1	ST1	001	ST2	010	1	0	1
L6	1	0	ST1	001	ST1	001	1	1	0
L6	1	1	ST1	001	ST1	001	1	1	0
L7	0	0	ST2	010	ST4	100	0	0	0
L8	0	1	ST2	010	ST4	100	0	0	1
L9	1	0	ST2	010	ST3	011	0	1	0
L9	1	1	ST2	010	ST3	011	0	1	0
L10	0	0	ST3	011	ST4	100	1	0	0
L11	0	1	ST3	011	ST4	110	1	0	1
L12	1	0	ST3	011	ST3	011	1	1	0
L12	1	1	ST3	011	ST3	011	1	1	0
L13	0	0	ST4	100	ST6	101	0	0	0
L14	0	1	ST4	100	ST6	110	0	0	1
L15	1	0	ST4	100	ST5	101	0	1	0
L15	1	1	ST4	100	ST5	101	0	1	0
L16	0	0	ST5	101	ST6	110	1	0	0
L17	0	1	ST5	101	ST6	110	1	0	1
L18	1	0	ST5	101	ST5	101	1	1	0

L18	1	1	ST5	101	ST5	101	1	1	0
L19	0	0	ST6	110	ST0	000	0	0	0
L20	0	1	ST6	110	ST0	000	0	0	1
L21	1	0	ST6	110	ST7	111	0	1	0
L21	1	1	ST6	110	ST7	111	0	1	0
L22	0	0	ST7	100	ST0	000	1	0	0
L23	0	1	ST7	100	ST0	000	1	0	1
L24	1	0	ST7	100	ST7	100	1	1	0
L24	1	1	ST7	100	ST7	100	1	1	0

5. BEVERAGE LEVEL SENSING AND REFILL CONTROL SYSTEM

The ASM chart for the lower tank beverage level sensing and refill control is shown in fig 7 along with its state map and state assignment. Before any production run, the control system of fig. 7 must check to see that the upper tank is not low. If low i.e. STOPRUN=1 it stops run and waits (i.e. WAITING) for upper tank refill. After the refill i.e. STOPRUN=0 the lower beverage tank levels (i.e. BVL) are sensed by the signal BVL1, BVL2, BVL3, BVL4 for beverage tanks 1 through 4 respectively. The tank is low on beverage when any of the signals BVL1 through BVL4 is logic 1 else the tank beverage level is high when logic 0. When any of the signals BVL1 BVL2, BVL3, BVL4 is low (meaning low on stock), the corresponding valve for that tank (HBV1, HBV2, HBV3, or HBV4) is turned on to let down the beverage from their corresponding upper tanks. This takes place at states ST1, ST3, ST5 and ST7 respectively. Meanwhile any time any of the valves (HBV1, HBV2, HBV3, HBV4) is turned on to let down beverage, from the upper tank, the level of the tank is monitored by the signals BVH1, BVH2, BVH3, BVH4 respectively to ascertain when it is filled. When BVHX=0 (where X= 1, 2, 3, 4) it means the beverage tank is not yet full hence it remains in the FBVX. If BVHX=1, it means that the lower tank in question is full



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

State map

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

State assignment

Fig -7: ASM chart for the lower tank level sensing and refill control

Note that this refill control system is applicable to any of the four columns of lower beverage tanks. Thus when one column's beverage tank sensing and refill control system is realized, it is replicated for use in the other columns. Table 3 shows the state agent oriented state transition table corresponding to the ASM chart of fig 7. The optimized ROM based implementation is shown in fig 8 through the use of a 1-out-of-8 multiplexer. The number of qualifier lines that reach the control system has been reduced to 1. The state agent based fully expanded transition table corresponding to this optimized ROM architecture is shown in table 4. Using the multi-agent (process agent plus state agents) based approach the full multi-agent based implementation of the beverage sensing and refill control system is shown in fig 9.

Table 3: state transition table corresponding to ASM chart of fig 7

State	Present State	Qualifiers									Next state	State outputs				
	C B A	STOPRUN	BVL1	BVH1	BVL2	BVH2	BVL3	BVH3	BVL4	BVH4	C′ B′ A′	WAITING	HBV1	HBV2	HBV3	HBV4
ST0	0 0 0	1	-	-	-	-	-	-	-	-	0 0 0	1	0	0	0	0
ST0	0 0 0	0	0	-	-	-	-	-	-	-	0 1 0	1	0	0	0	0
ST0	0 0 0	0	1	-	-	-	-	-	-	-	0 0 1	1	0	0	0	0
ST1	0 0 1	-	-	0	-	-	-	-	-	-	0 0 1	0	1	0	0	0
ST1	0 0 0	-	-	1	-	-	-	-	-	-	0 1 0	0	1	0	0	0
ST2	0 1 0	-	-	-	0	-	-	-	-	-	1 0 0	0	0	0	0	0
ST2	0 1 0	-	-	-	1	-	-	-	-	-	0 1 1	0	0	0	0	0
ST3	0 1 1	-	-	-	-	0	-	-	-	-	0 1 1	0	0	1	0	0
ST3	1 0 0	-	-	-	-	1	-	-	-	-	1 0 0	0	0	1	0	0
ST4	1 0 0	-	-	-	-	-	0	-	-	-	1 1 0	0	0	0	0	0
ST4	1 0 0	-	-	-	-	-	1	-	-	-	1 0 1	0	0	0	0	0
ST5	1 0 1	-	-	-	-	-	-	0	-	-	1 0 1	0	0	0	1	0
ST5	1 0 1	-	-	-	-	-	-	1	-	-	1 1 0	0	0	0	1	0
ST6	1 1 0	-	-	-	-	-	-	-	0	-	0 0 0	0	0	0	0	0
ST6	1 1 0	-	-	-	-	-	-	-	1	-	1 1 1	0	0	0	0	0
ST7	1 1 1	-	-	-	-	-	-	-	-	0	1 1 1	0	0	0	0	1
ST7	1 1 1	-	-	-	-	-	-	-	-	1	0 0 0	0	0	0	0	1

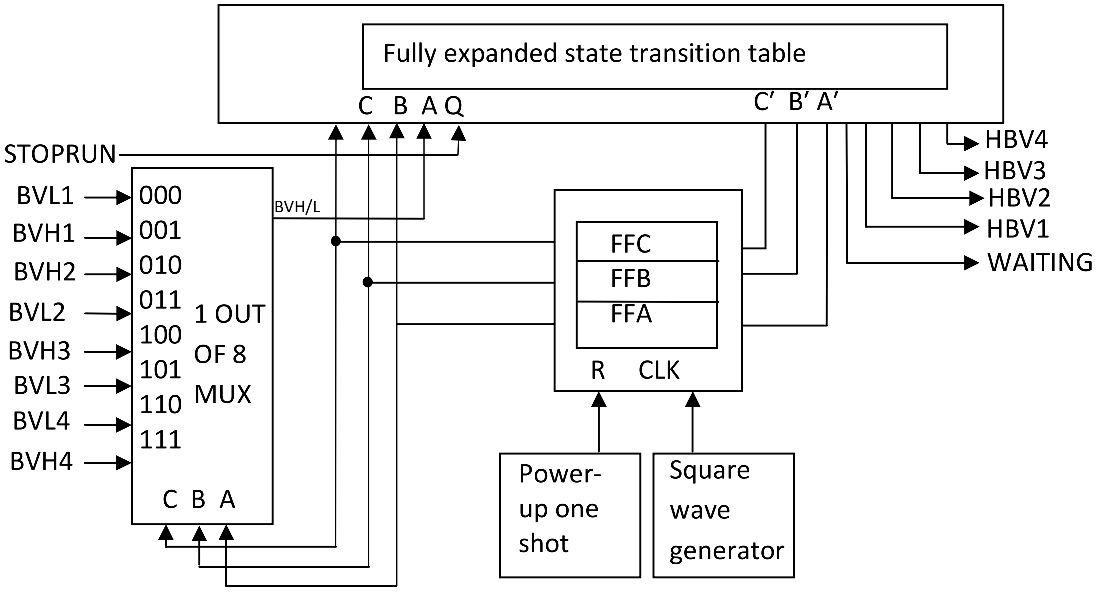


Fig-8: ROM-based implementation of sensing and refill control system

Table -4: fully expanded agent based table corresponding fig 8

State	Present State			Qualifiers		Next State			State outputs					HEX
	C	B	A	STOPRUN	Q	C′	B′	A′	WAITING	HBV1	HBV2	HBV3	HBV4	OUT-PUT
ST0	0	0	0	0	0	0	1	0	1	0	0	0	0	50
	0	0	0	0	1	0	0	1	1	0	0	0	0	30
	0	0	0	1	0	0	0	0	1	0	0	0	0	10
	0	0	0	1	1	0	0	0	1	0	0	0	0	10
ST1	0	0	1	0	0	0	0	1	0	1	0	0	0	28
	0	0	1	0	1	0	1	0	0	1	0	0	0	48
	0	0	1	1	0	0	0	1	0	1	0	0	0	28
	0	0	1	1	1	0	1	0	0	1	0	0	0	48
ST2	0	1	0	0	0	1	0	0	0	0	0	0	0	40
	0	1	0	1	0	0	1	1	0	0	0	0	0	60
	0	1	0	0	1	1	0	0	0	0	0	0	0	40
	0	1	0	1	1	0	1	1	0	0	0	0	0	60
ST3	0	1	1	0	0	0	1	1	0	0	1	0	0	64
	0	1	1	0	1	1	0	0	0	0	1	0	0	84
	0	1	1	1	0	0	1	1	0	0	1	0	0	64
	0	1	1	1	1	1	0	0	0	0	1	0	0	84
ST4	1	0	0	0	0	1	1	0	0	0	0	0	0	C0
	1	0	0	0	1	1	0	1	0	0	0	0	0	C0
	1	0	0	1	0	1	1	0	0	0	0	0	0	A0
	1	0	0	1	1	1	0	1	0	0	0	0	0	A0
ST5	1	0	1	0	0	1	0	1	0	0	0	1	0	A2
	1	0	1	0	1	1	1	0	0	0	0	1	0	A2
	1	0	1	1	0	1	0	1	0	0	0	1	0	C2
	1	0	1	1	1	1	1	0	0	0	0	1	0	C2
ST6	1	1	0	0	0	0	0	1	0	0	0	0	0	00
	1	1	0	0	1	1	1	0	0	0	0	0	0	00
	1	1	0	1	0	0	0	1	0	0	0	0	0	E0
	1	1	0	1	1	1	1	0	0	0	0	0	0	E0
ST7	1	1	1	0	0	1	1	1	0	0	0	0	1	E1
	1	1	1	0	1	0	0	0	0	0	0	0	1	E1
	1	1	1	1	0	1	1	1	0	0	0	0	1	01
	1	1	1	1	1	0	0	0	0	0	0	0	1	01

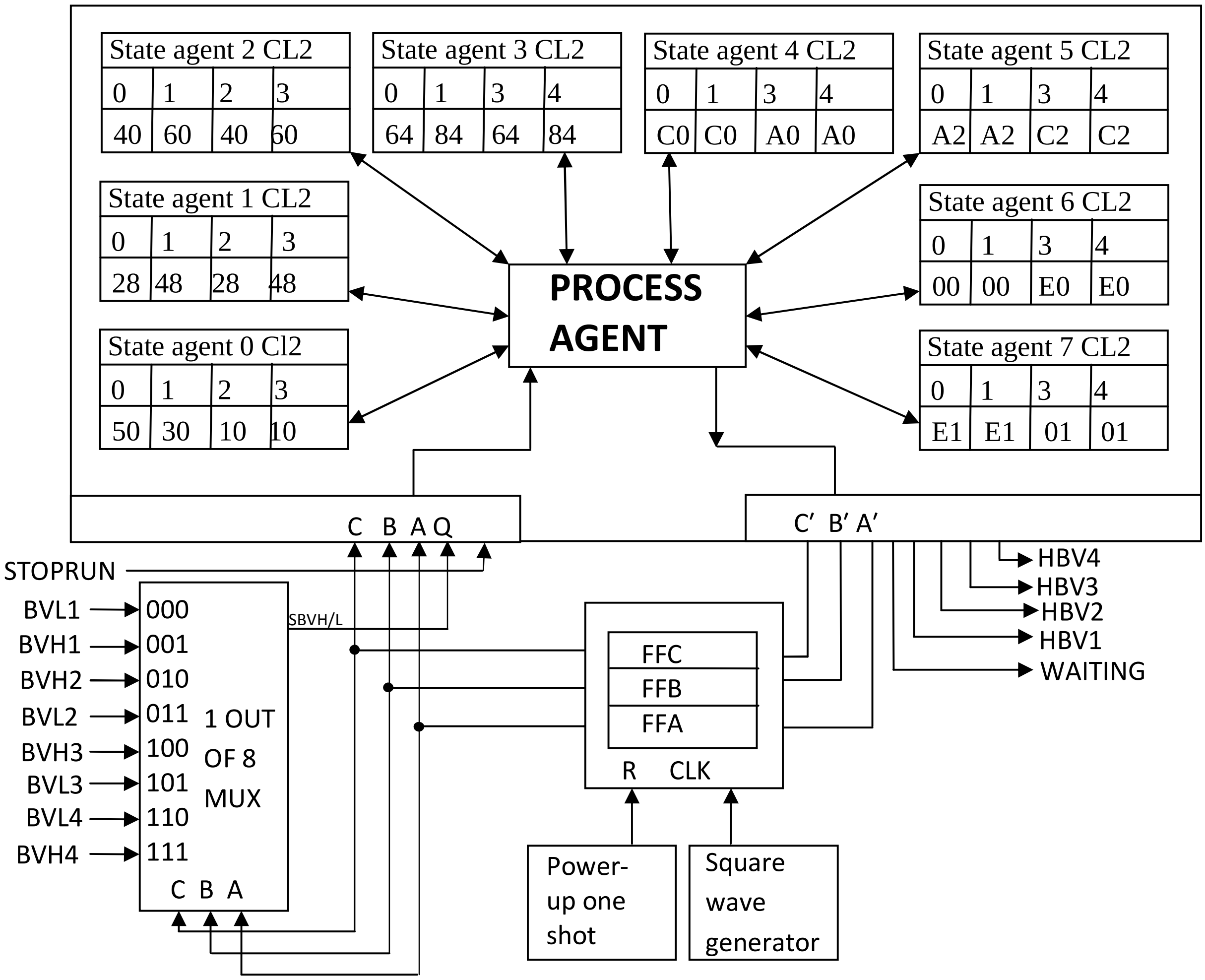
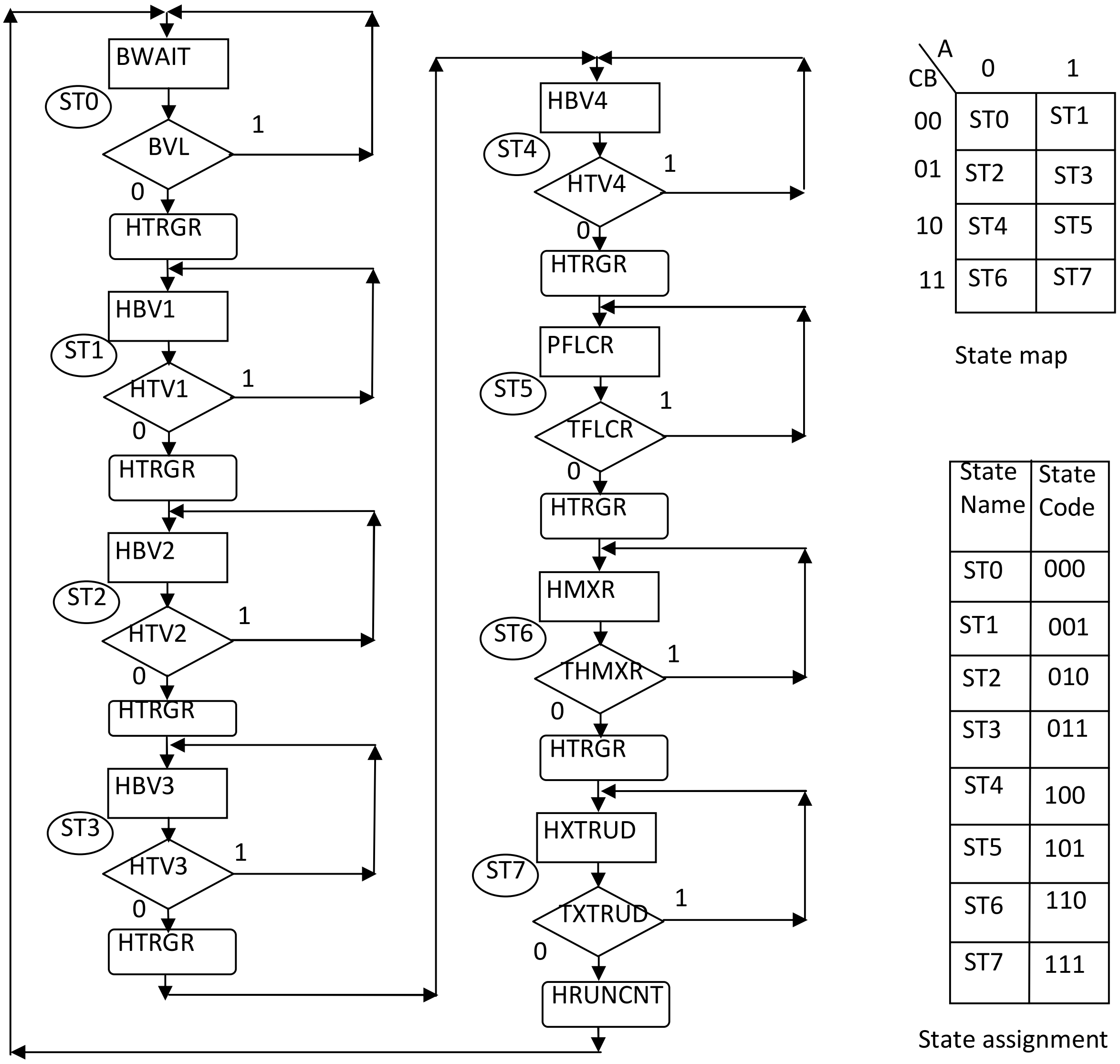


Fig- 9: Multi-Agent based implementation of sensing and refill control system

6. THE INDUSTRIAL BLENDING PROCESS

The ASM chart of fig 10 shows the algorithm for the blending process that takes place at each of the mixers 1, 2, 3 & 4 along with its state map and state assignment.

The ASM chart of fig 4.10 has 8 states. The first state with the signal 'BWAIT' indicates that the beverage blender is waiting for any of the tanks that is low to be refilled. It should be noted that if any of the upper tanks is low, or any of the lower tanks is low (i.e. BVL=1), then the beverage blending control system must remain in the waiting state i.e. BWAIT. When it is refilled (BVL=0), the valve for tank 1(HBV1) is opened to dispense beverage 1 and simultaneously a timer for the valve (HTV1) is triggered (by HTRGR). Once time timing is up it moves to trigger the timing and dispensing of beverage 2, then beverage 3 and beverage 4 in that order. After this the predetermined quantity of flavor and colour is added at state ST5 (i.e. PFLCR). When this is done (i.e. TFLCR=0) the beverages are blended/mixed (HMXR) which is timed by TMXR. At TMXR=0 the beverage is extruded (i.e. XTRUD) when this is done (i.e. at TXTRUD=0) the run count signal is triggered (HRUNCNT) and then the control system waits (BWAIT) for the next production run. Thus, when this beverage blending control system is realized for one mixer, it is replicated for use with other columns of lower beverage tanks & mixer.



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

State map

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

State assignment

Fig 10: ASM for beverage blending process

Table 5 shows the state agent oriented state transition table corresponding to the ASM chart of fig 10. The optimized ROM based implementation of the ASM chart is shown in fig 11. Through the use of a 1-out-of -8 multiplexer, the number of qualifier lines that is connected directly to the control system has been reduced to 1 namely Q. The state agent based fully expanded transition table corresponding to the optimized ROM implementation of fig 11 is shown in table 6. Again using the multi-agent approach (i.e. state agents controlled by a process agent) the implementation of the industrial blending process control is as shown in fig 12. It features the use of a decoder, as described by Tocci et al (2009), to decode state outputs so as to limit the output line demand of this state machine to 1 byte or less as required in the optimized ROM design approach showed by Inyiama et al (in print).

Table 5: Agent- STT corresponding to ASM of fig. 10

STATE	PRESENT STATE	QUALIFIERS								NEXT STATE	STATE OUTPUTS								CONDITIONAL OUTPUT	
	C B A	BVL	HTV1	HTV2	HTV3	HTV4	PFLCR	TMXR	TXTRUD	C′ B′ A′	BWAIT	HBV1	HBV2	HBV3	HBV4	HFLCR	HMXR	HXTRUD	HTRGR	HRUNCNT
STO	000	1	-	-	-	-	-	-	-	000	1	0	0	0	0	0	0	0	0	0
STO	000	0	-	-	-	-	-	-	-	001	1	0	0	0	0	0	0	0	1	0
ST1	001	-	1	-	-	-	-	-	-	001	0	1	0	0	0	0	0	0	0	0
ST1	001	-	0	-	-	-	-	-	-	010	0	1	0	0	0	0	0	0	1	0
ST2	010	-	-	1	-	-	-	-	-	010	0	0	1	0	0	0	0	0	0	0
ST2	010	-	-	0	-	-	-	-	-	011	0	0	1	0	0	0	0	0	1	0
ST3	011	-	-	-	1	-	-	-	-	011	0	0	0	1	0	0	0	0	0	0
ST3	011	-	-	-	0	-	-	-	-	100	0	0	0	1	0	0	0	0	1	0
ST4	100	-	-	-	-	1	-	-	-	100	1	0	0	0	1	0	0	0	0	0
ST4	100	-	-	-	-	0	-	-	-	101	1	0	0	0	1	0	0	0	1	0
ST5	101	-	-	-	-	-	1	-	-	101	1	0	0	0	0	1	0	0	0	0
ST5	101	-	-	-	-	-	0	-	-	110	1	0	0	0	0	1	0	0	1	0
ST6	110	-	-	-	-	-	-	1	-	110	1	0	0	0	0	0	1	0	0	0
ST6	110	-	-	-	-	-	-	0	-	111	1	0	0	0	0	0	1	0	1	0
ST7	111	-	-	-	-	-	-	-	1	111	1	0	0	0	0	0	0	1	0	0
ST7	111	-	-	-	-	-	-	-	0	000	1	0	0	0	0	0	0	1	0	1

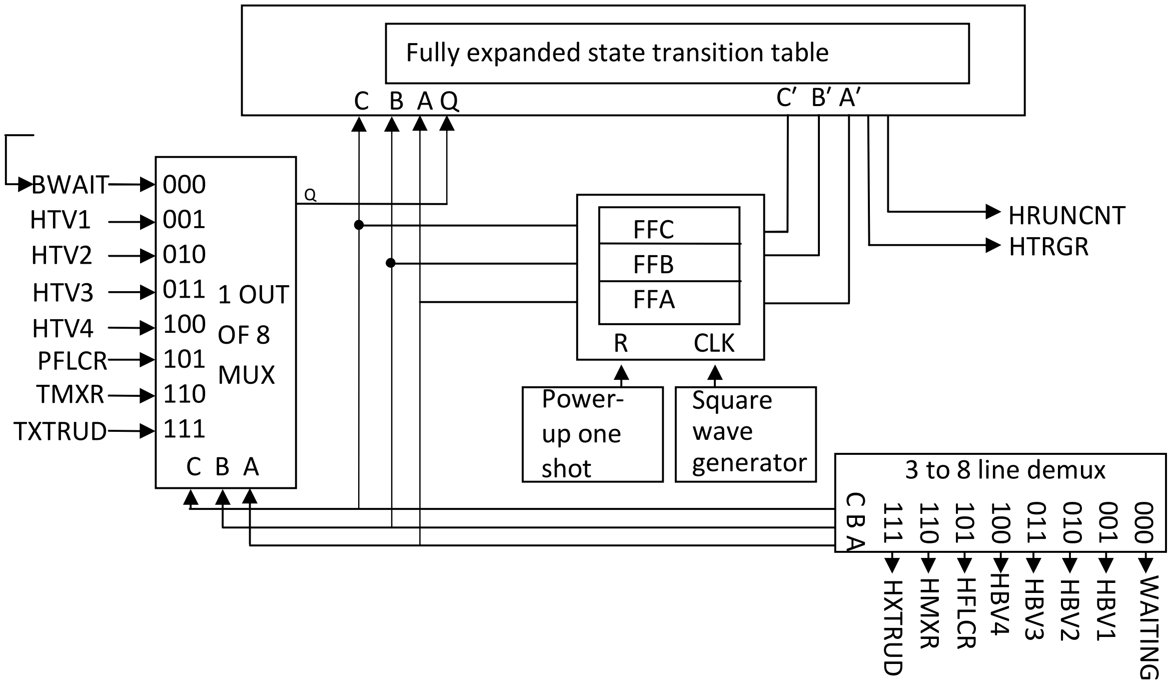


Fig- 11: Optimized ROM-based implementation of beverage blending/mixing control system

Table 6: Agent-based transition table derived from fig. 11

STATE	PRESENT STATE	QUALIFIERS (Q)	NEXT STATE	CONDITIONAL OUTPUT		OUTPUT IN HEX
	C B A		C′ B′ A′	HTRGR	HRUNCNT	
STO	000	1	000	0	0	00
STO	000	0	001	1	0	06
ST1	001	1	001	0	0	04
ST1	001	0	010	1	0	06
ST2	010	1	010	0	0	08
ST2	010	0	011	1	0	0D
ST3	011	1	011	0	0	0C
ST3	011	0	100	1	0	12
ST4	100	1	100	0	0	10
ST4	100	0	101	1	0	16
ST5	101	1	101	0	0	14
ST5	101	0	110	1	0	1A
ST6	110	1	110	0	0	18
ST6	110	0	111	1	0	1E
ST7	111	1	111	0	0	1C
ST7	111	0	000	0	1	01

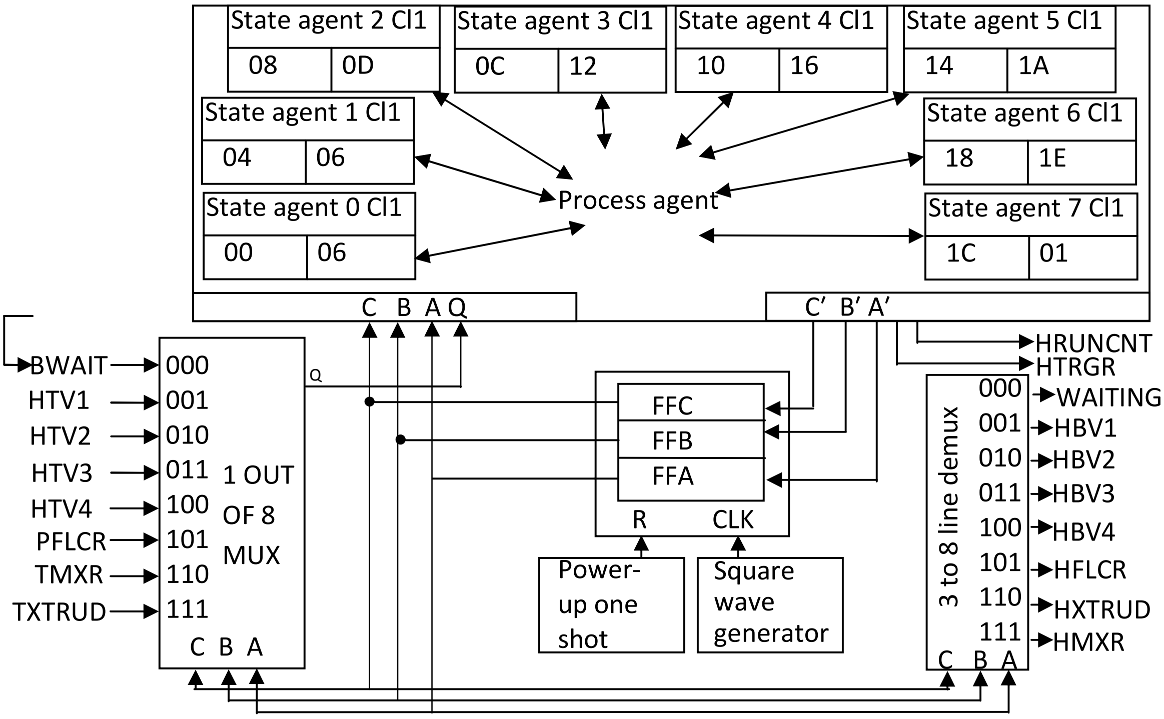


Fig- 12: multi-agent-based implementation of beverage blending/mixing control system

Fig 13 shows the multi-agent based linked state machine for the industrial blender. It comprises of three processes namely: upper tank control, the beverage level sensing and refill control and the beverage blending operation.

Each unit is working independently but the upper tank control passes a signal to the lower tank control when it runs short of beverage. Similarly, the lower tank control sends a signal 'BWAIT' to the blending agent when it is waiting to refill the tanks that have gone low on stock.

The state agents of each of the state machines do not have communication with those of other units. It is the process agents that do the communication between the state machines

.

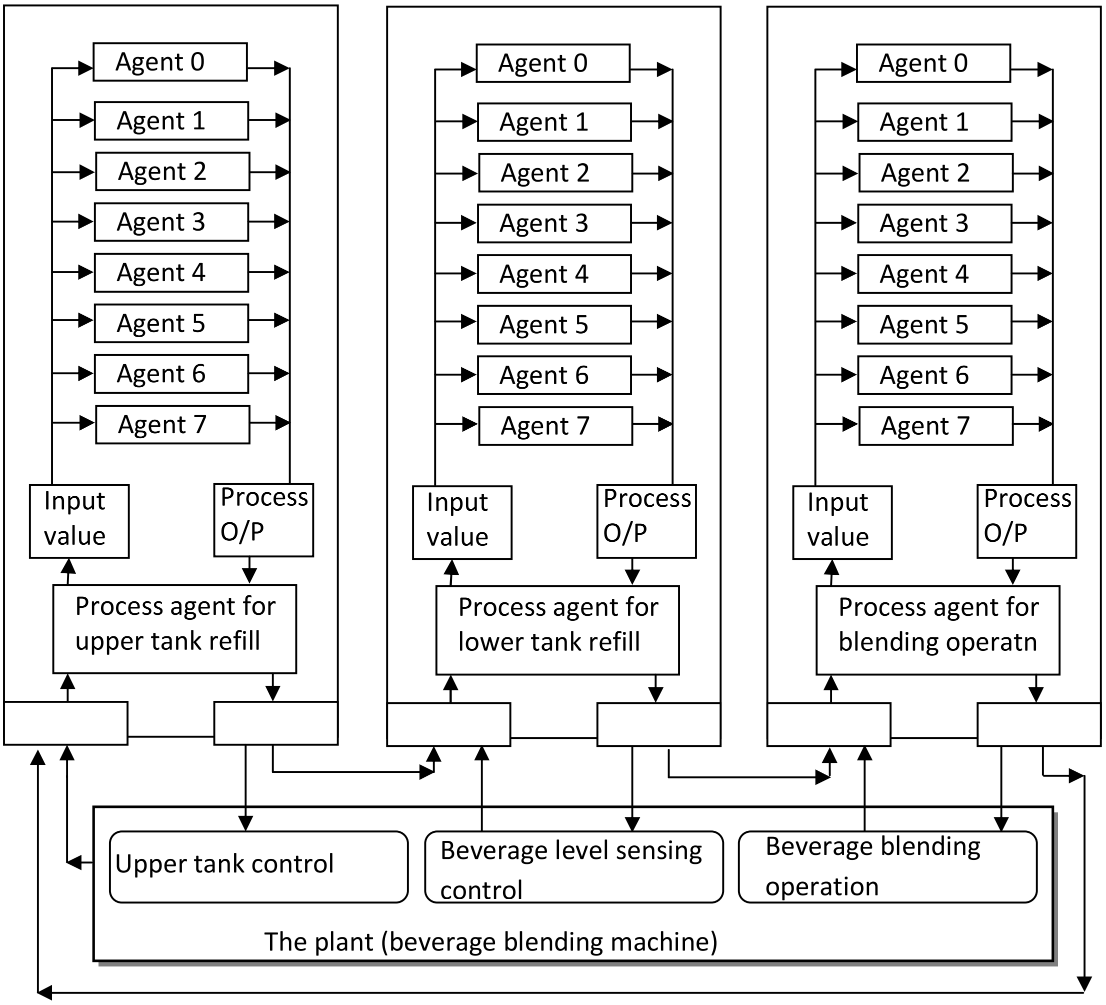

Agent 0
Agent 1
Agent 2
Agent 3
Agent 4
Agent 5
Agent 6
Agent 7
Input value
Process O/P
Process agent for upper tank refill
Process agent for lower tank refill
Process agent for blending operatn
Upper tank control
Beverage level sensing control
Beverage blending operation
The plant (beverage blending machine)

Fig 13: Multi-agent based linked state machine

CONCLUSIONS

A systematic approach to the design of multi-agent based linked state machines has been demonstrated in the foregoing. The method reduces the effort required to realize complex process control systems through the 'divide –and-conquer approach. The complex process control system is first broken down into a number of simpler control systems which are quickly realized and then linked up so as to yet constitute the complex control system. In the multi-agent based approach, each of the simpler process control systems is implemented as a multi-agent based system comprised of one process agent and a number of state agents. Only the process agent in each state machine is used to perform the intelligent interlinking of all the machines in order to realize a composite complex process control system built by concatenating a number of simpler systems.

REFERENCES

[1]. Akpado K. A., Inyiama H. C., Okezie C.C. (2010), "Simplifying industrial process control using linked state machine approach". *International Journal of Electrical and Telecommunication System Research.* Vol 4. No 4 pp 72-79.

[2]. Inyiama H.C., C.C. Okezie and I.C. Okafo (2011), "Digital Control of Palm Fruit Processing using ROM Based Linked State Machines" *European Journal of Scientific Research,*Vol.59 No.4 Pp. 597-606.

[3]. Inyiama H.C., C.C. Okezie and I.C. Okafo (2012), "Agent based process control system design". *Proceedings of the peer reviewed National Conference on Infrastructural Development and Maintenance in the Nigerian Environment.* Pp 234-252

[4]. Inyiama H.C., C.C. Okezie and I.C. Okafo (In print), "Complexity Reduction in Rom-Based Process Control Systems Via Input Multiplexing and Output Decoding" *International Journal Of Engineering Innovations.* Paper No: AJUS-2011-115

[5]. Tocci R.J, N.S Widmer, G.L Moss, (2009), *Digital systems principles and applications* 10th ed; Pearson; Pp 435.