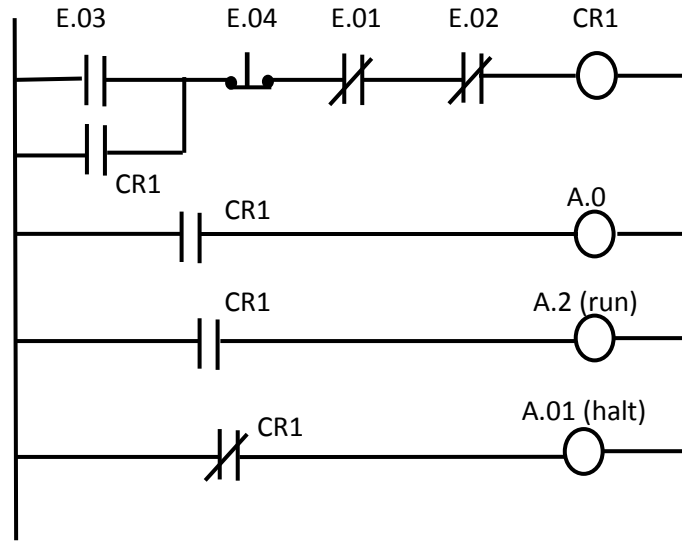


1. Bollinger 5.16

(a)



(b) Program driven synchronization

CLOCK	EQUALS	E.0
LS1	EQUALS	E.1
LS2	EQUALS	E.2
START	EQUALS	E.3
STOP	EQUALS	E.4
FBACK	EQUALS	E.5
MANIP	EQUALS	A.0
HALT	EQUALS	A.1
RUN	EQUALS	A.2
C	EQUALS	B.1
R	EQUALS	B.0
K	EQUALS	B.2

BEGIN	LOAD	#0	(Default state of the system, optional)
-------	------	----	---

	STORE	MANIP
--	-------	-------

	LOAD	#1
--	------	----

	STORE	HALT
--	-------	------

	LOAD	#0
--	------	----

	STORE	RUN
--	-------	-----

WAITS	INPUT	START	(Starting Process)
-------	-------	-------	--------------------

	JUMP IF ZERO	WAITS
--	--------------	-------

	LOAD	#0
--	------	----

	STORE	HALT
--	-------	------

	LOAD	#1
--	------	----

	STORE	RUN
--	-------	-----

LOOP	INPUT	STOP	(Stopping process)
------	-------	------	--------------------

	JUMP IF NOT ZERO	BEGIN
--	------------------	-------

	INPUT	LS1
--	-------	-----

	JUMP IF NOT ZERO	BEGIN
--	------------------	-------

	INPUT	LS2
--	-------	-----

	JUMP IF NOT ZERO	BEGIN
--	------------------	-------

WAIT0	INPUT	CLOCK	(Feedback control implementation)
-------	-------	-------	-----------------------------------

	JUMP IF NOT ZERO	WAIT0
--	------------------	-------

WAIT1	INPUT	CLOCK
-------	-------	-------

	JUMP IF NOT ZERO	WAIT1
--	------------------	-------

	INPUT	FBACK
--	-------	-------

	STORE	C
--	-------	---

	LOAD	R
--	------	---

	SUBTRACT	C
--	----------	---

	MULTIPLY	K
--	----------	---

	OUTUT	MANIP
--	-------	-------

	JUMP	LOOP
--	------	------

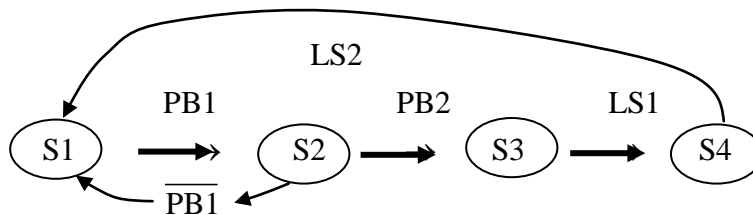
2.

$$\begin{aligned} & A \cdot B + \overline{A \cdot C} + C \cdot \overline{D} + \overline{B} \cdot \overline{D} \\ &= \underbrace{A \cdot B + \overline{A}}_{\text{red arrow}} + \overline{C} + C \cdot \overline{D} + \overline{B} \cdot \overline{D} \\ &= \underbrace{A \cdot B + (1+B) \cdot \overline{A}}_{\text{red arrow}} + \overline{C} + C \cdot \overline{D} + \overline{B} \cdot \overline{D} \\ &= \overline{A} + B + \overline{C} + C \cdot \overline{D} + \overline{B} \cdot \overline{D} \\ &= \overline{A} + \overline{C} + \overline{D} + B + \overline{B} \cdot \overline{D} \\ &= \overline{A} + \overline{C} + \overline{D} + B + \overline{D} \\ &= \overline{A} + B + \overline{C} + \overline{D} \end{aligned}$$

$$\begin{aligned} & (A+B) \cdot (A+\overline{B}+C) \\ &= (A+B) \cdot A + A \cdot \overline{B} + B \cdot \overline{B} + (A+B) \cdot C \\ &= A + A \cdot \overline{B} + (A+B) \cdot C \\ &= A + (A+B) \cdot C \\ &= A + A \cdot C + B \cdot C \\ &= A + B \cdot C \end{aligned}$$

### 3. Bollinger 9.18

One way to automate the machine is to leave steps 1 and 2 as it is, but add limit switches to automate steps 3 and 4 so that the operator can tend another machine while the saw in advancing and retracting. Limit switch LS1 should be placed so that it is activated when the saw reaches the desired final position. Limit switch LS2 should be placed so that it is activated when the saw is in a fully retracted position. (There could be other variations)



Sol1=0	Sol1=1	Sol1=1	Sol1=1
Sol2=0	Sol2=0	Sol2=1	Sol2=1
Sol3=0	Sol3=0	Sol3=1	Sol3=0

$$\begin{aligned}
 S1 &= (S1 + LS2 + \overline{PB1}) \cdot \overline{PB1} \\
 &= S1 \cdot \overline{PB1} + LS2 \cdot \overline{PB1} + \overline{PB1} = (S1 + 1) \cdot \overline{PB1} + LS2 \cdot \overline{PB1} \\
 &= \overline{PB1} + LS2 \cdot \overline{PB1} = (1 + LS2) \cdot \overline{PB1} = \overline{PB1}
 \end{aligned}$$

$$\begin{aligned}
 S2 &= (S2 + PB1) \cdot (\overline{PB2} + \overline{PB1}) \\
 &= (S2 + PB1) \cdot (\overline{PB2} \cdot \overline{PB1}) \\
 &= S2 \cdot \overline{PB1} \cdot \overline{PB2} + PB1 \cdot \overline{PB2} \\
 &= (S2 + 1) \cdot \overline{PB1} \cdot \overline{PB2} = \overline{PB1} \cdot \overline{PB2}
 \end{aligned}$$

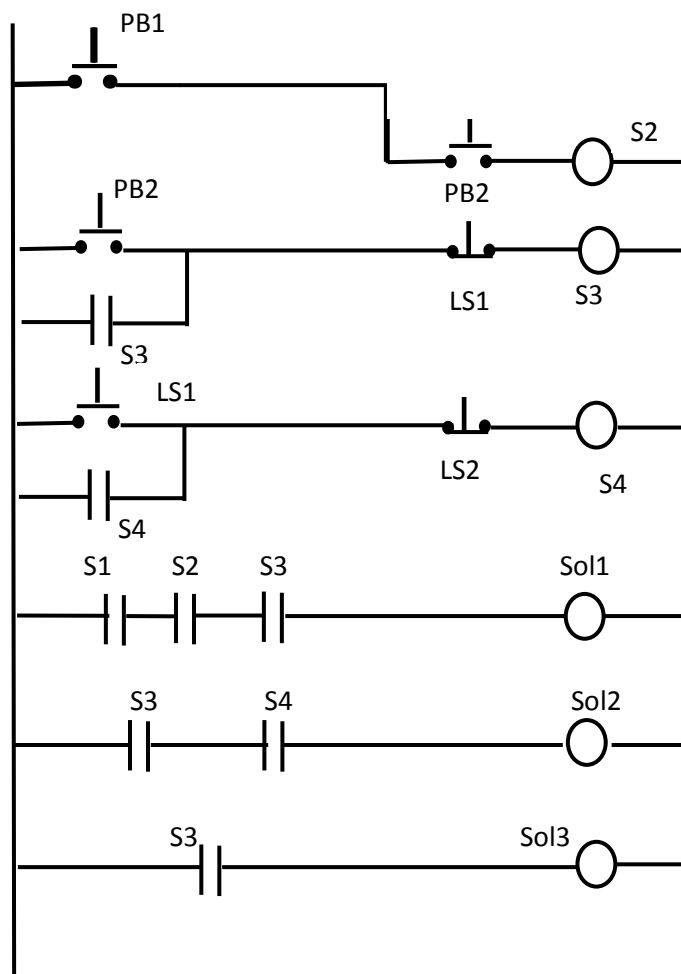
$$S3 = (S3 + PB2) \cdot \overline{LS1}$$

$$S4 = (S4 + LS1) \cdot LS2$$

$$SOL1 = S2 + S3 + S4$$

$$SOL2 = S3 + S4$$

$$SOL3 = S3$$



Alternative design (simpler)

