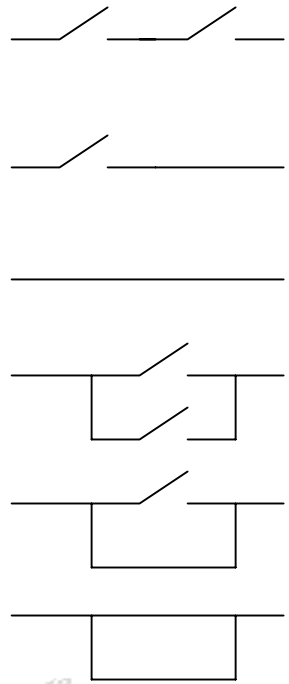
Boolean Arithmetic

- 0 and 0 = 0 (0 • 0 = 0)
- 0 and 1 = 0 (0 • 1 = 0)
- 1 and 1 = 1 (1 • 1 = 1)
- 0 or 0 = 0 (0 + 0 = 0)
- 0 or 1 = 1 (0 + 1 = 1)
- 1 or 1 = 1 (1 + 1 = 1)
- not 0 = 1 (/0 = 1)



A • /A = 1

A •B = B • A

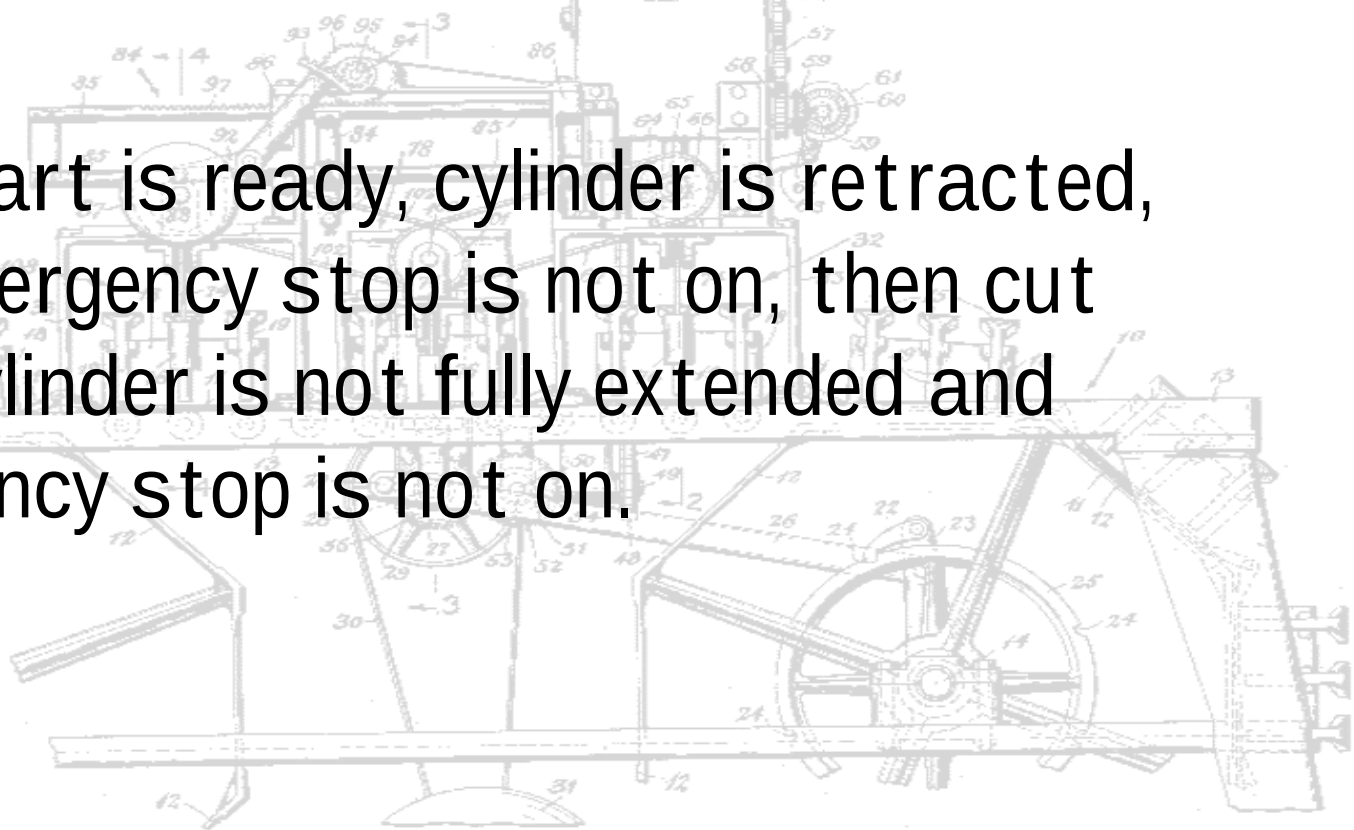
A • B + A • C = A • (B+C)

Digital Logic Expressions

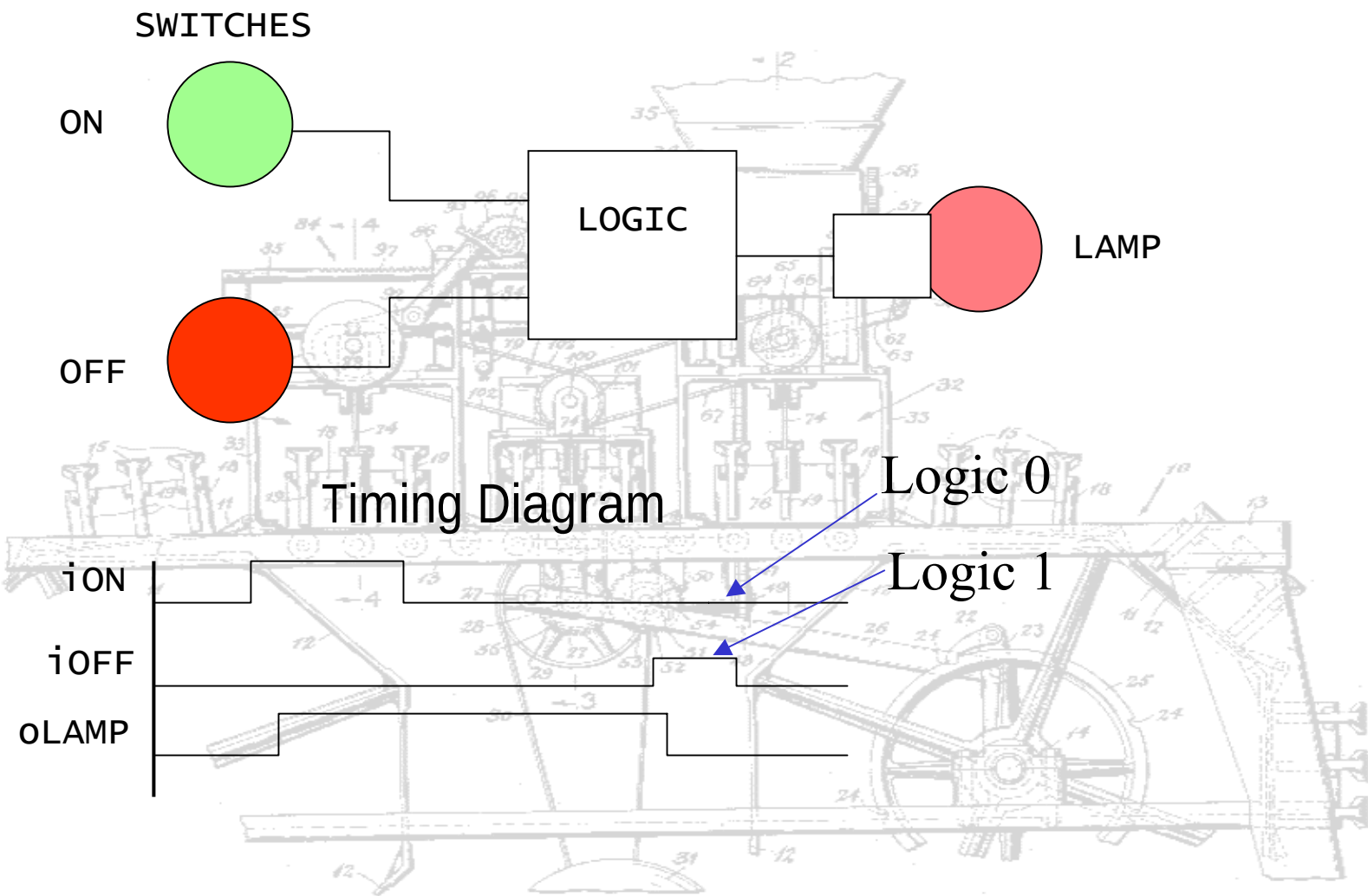
```
oCUT = (iPARTRDY•iCYLRETRACT
          + oCUT•/iCYLEXTEND)•/iESTOP
```

```
oCUT = iPARTRDY•iCYLRETRACT•/iESTOP
          + oCUT•/iCYLEXTEND•/iESTOP
```

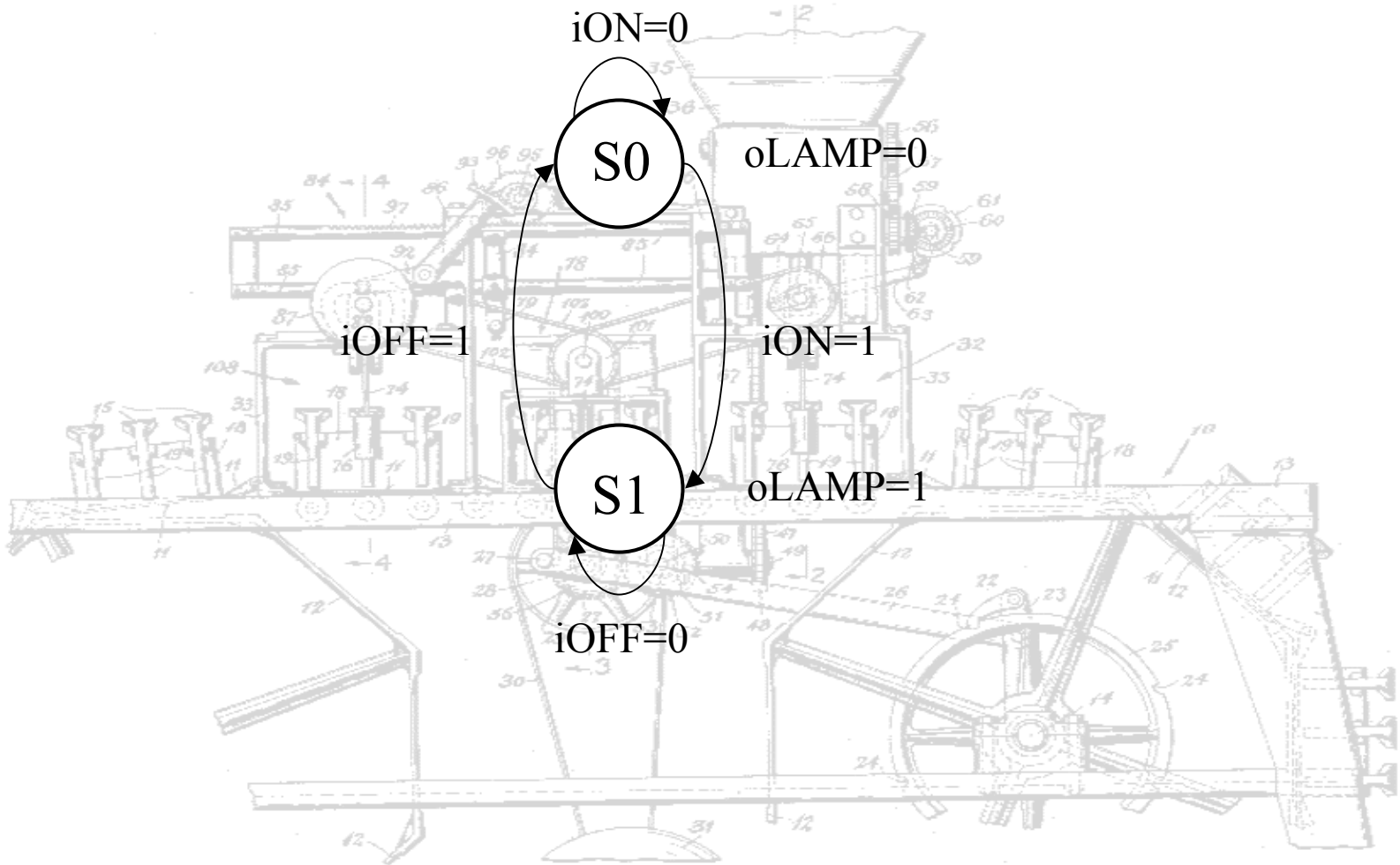
When part is ready, cylinder is retracted, and emergency stop is not on, then cut while cylinder is not fully extended and emergency stop is not on.



The Simplest State Machine

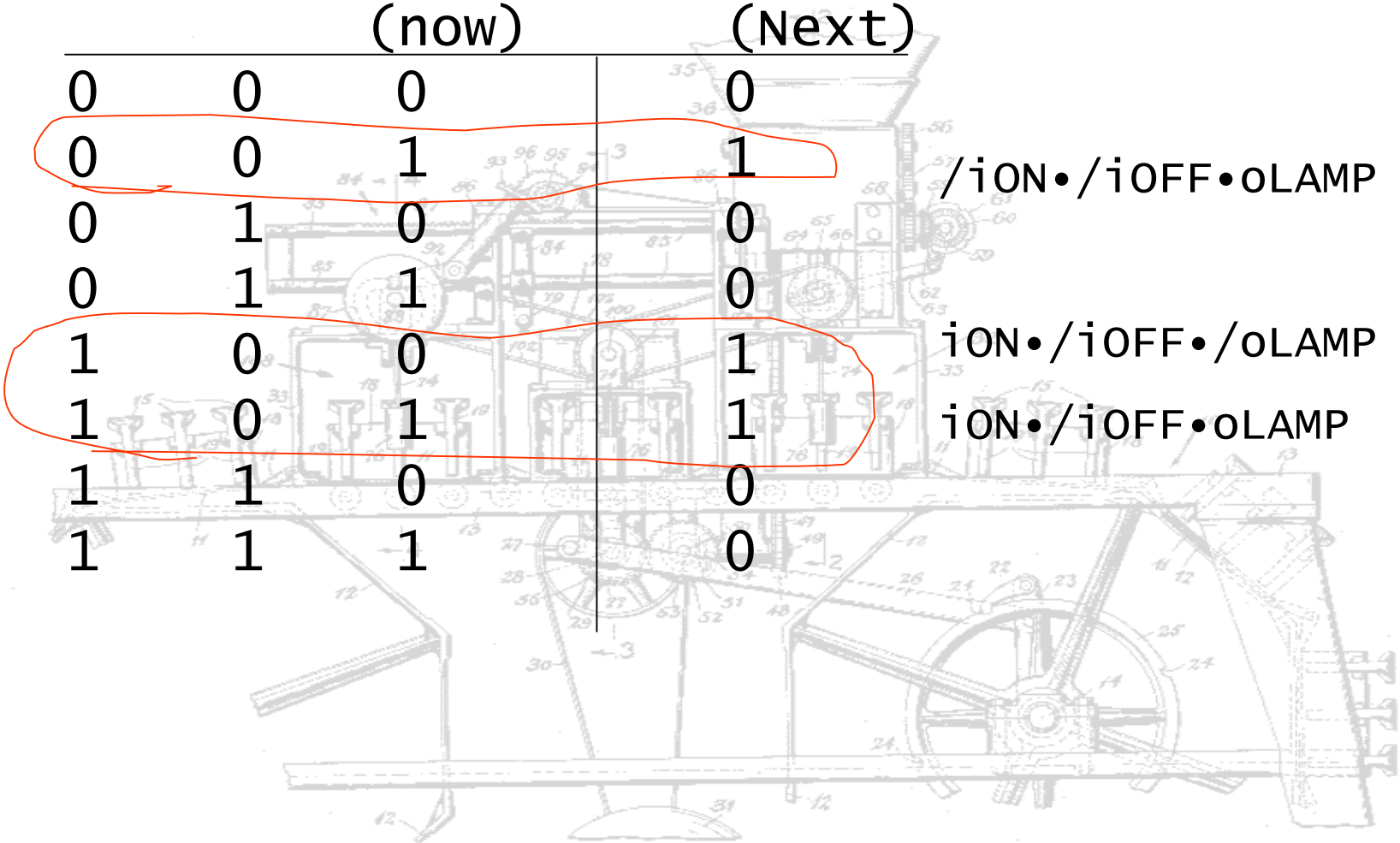


The Simplest State Machine



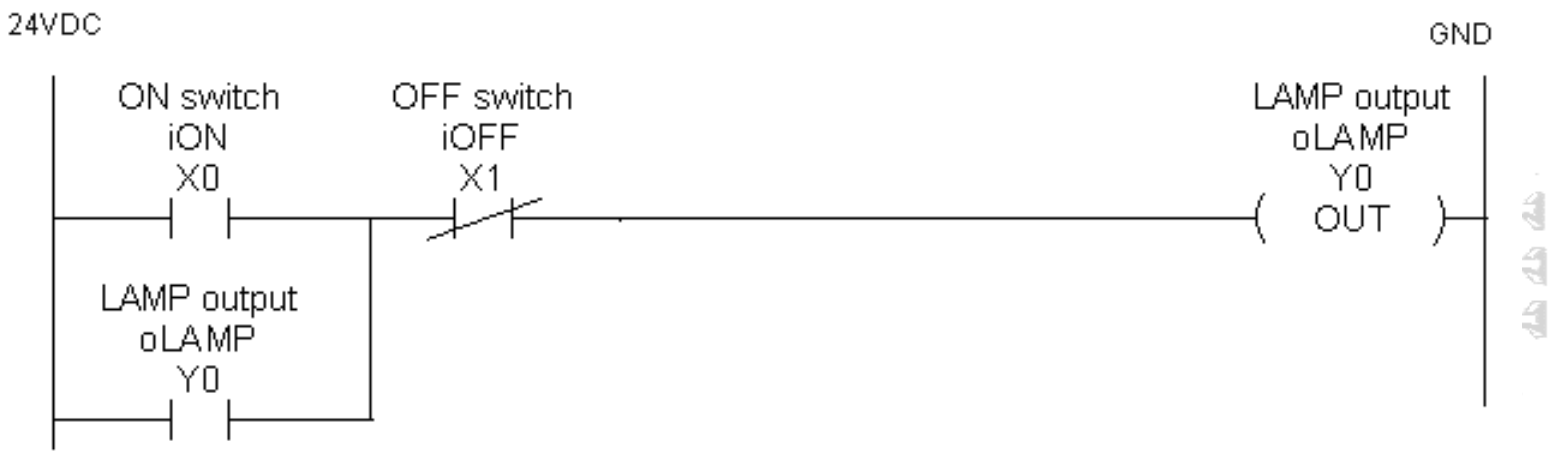
State Transition Table

iON	iOFF	oLAMP (now)	oLAMP (Next)	
0	0	0	0	
0	0	1	1	/iON•/iOFF•oLAMP
0	1	0	0	
0	1	1	0	
1	0	0	1	iON•/iOFF•/oLAMP
1	0	1	1	iON•/iOFF•oLAMP
1	1	0	0	
1	1	1	0	

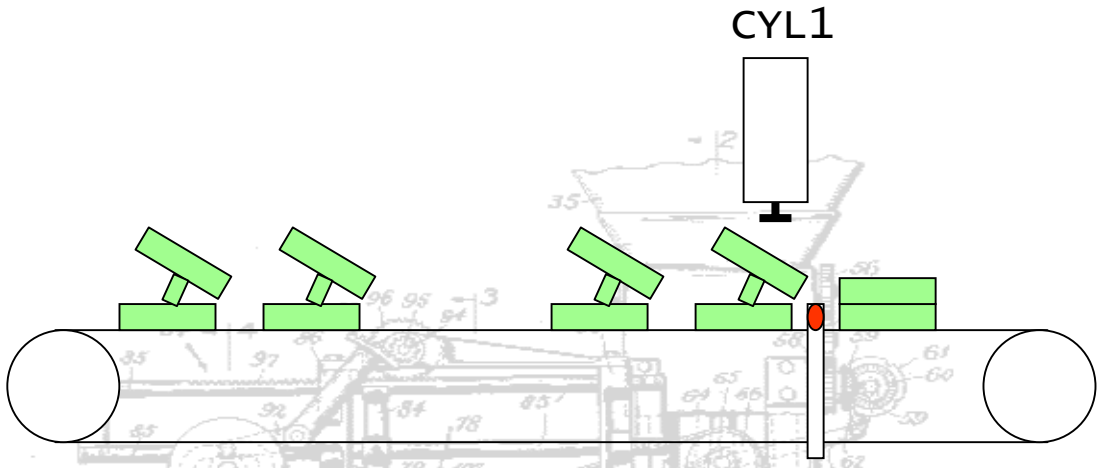


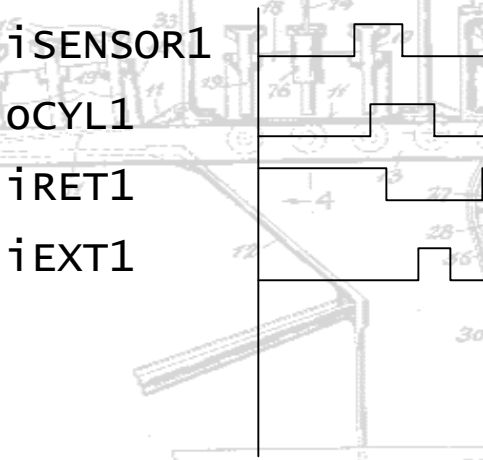
Logic Eq. & Ladder Diagram

$$
\begin{aligned}
oLAMP &= /iON \bullet /iOFF \bullet oLAMP \\
&\quad + iON \bullet /iOFF \bullet /oLAMP \\
&\quad + iON \bullet /iOFF \bullet oLAMP \\
&= iON \bullet /iOFF \bullet (/oLAMP + oLAMP) \\
&\quad + /iOFF \bullet oLAMP \bullet (/iON + iON) \\
&= iON \bullet /iOFF + /iOFF \bullet oLAMP \\
&= /iOFF \bullet (iON + oLAMP)
\end{aligned}
$$



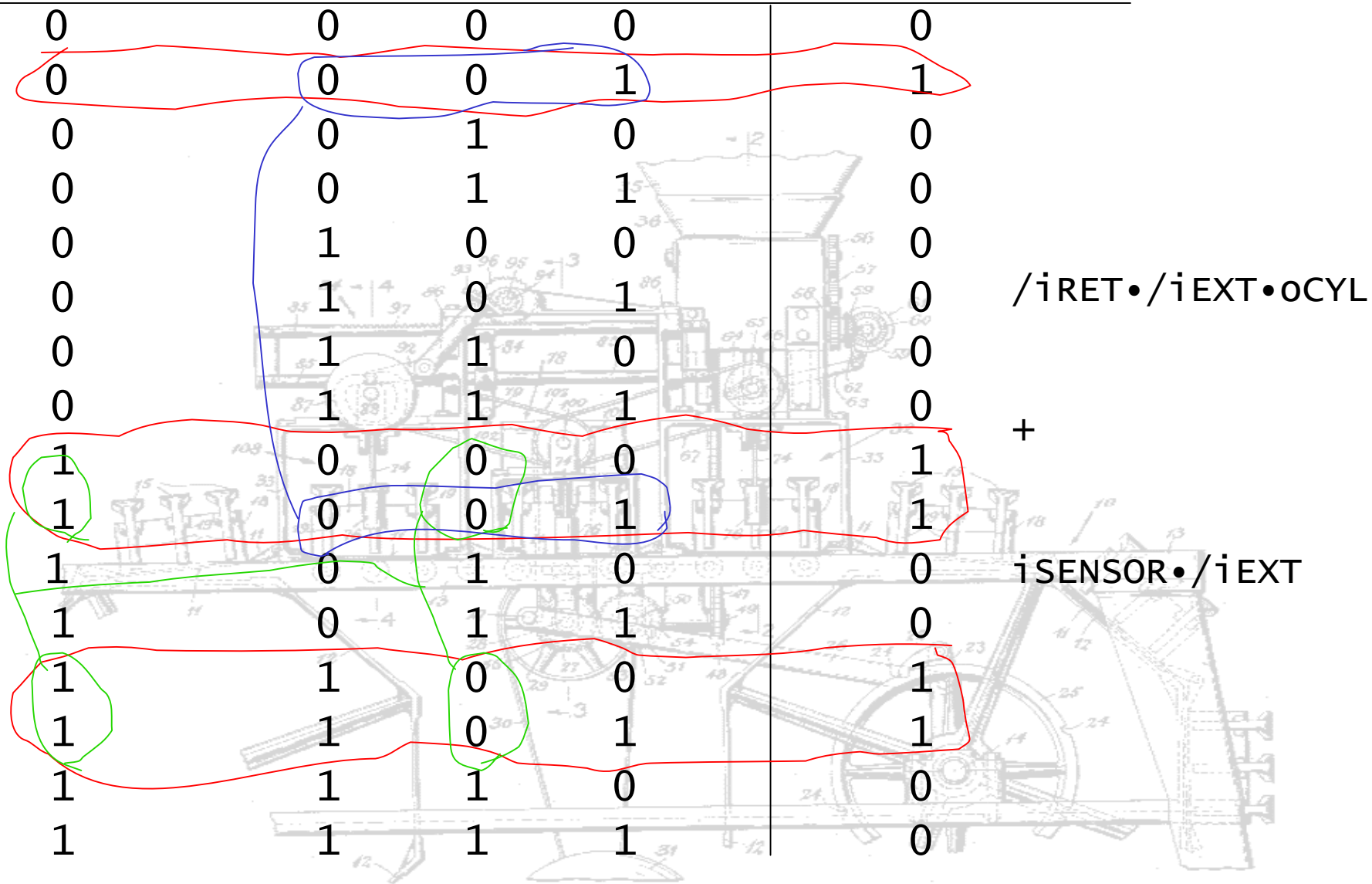
Another Example





Example #2

iSENSOR	iRET	iEXT	oCYL(curr)	oCYL (next)	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	0	/iRET•/iEXT•oCYL
0	1	1	0	0	
0	1	1	1	0	+
1	0	0	0	1	
1	0	0	1	1	
1	0	1	0	0	iSENSOR•/iEXT
1	0	1	1	0	
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	0	



Example#2 Ladder Logic

$$oCYL = iSENSOR \cdot /iRET$$
$$+ /iRET \cdot /iEXT \cdot oCYL$$
$$= (iSENSOR + /iEXT \cdot oCYL) \cdot /iRET$$