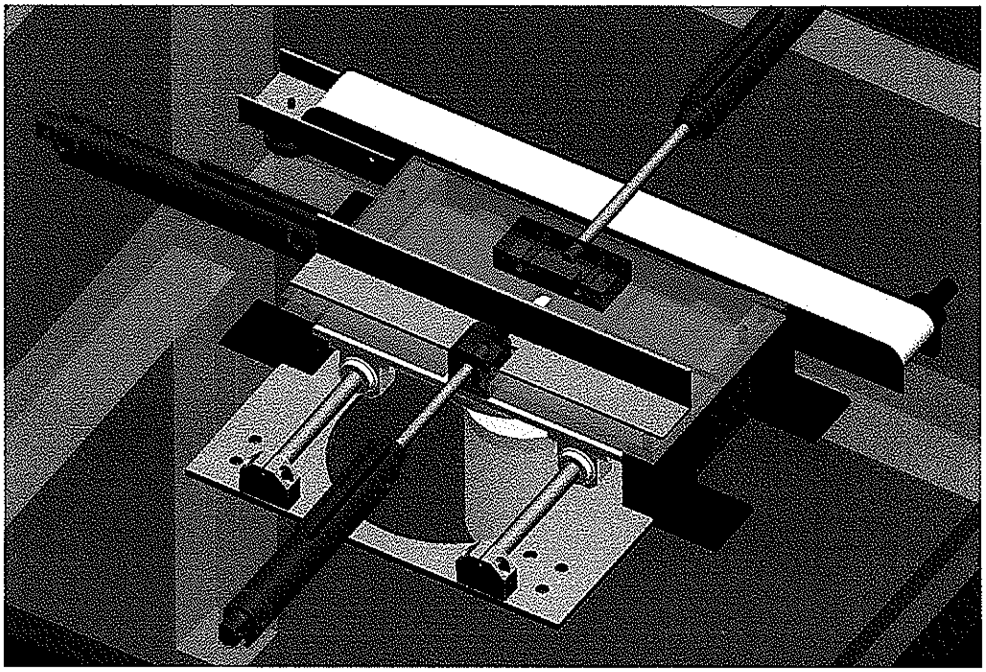
Awesome-O Burr Puzzle Piece Production System



Matt Pawluk

Kevin Christman

Ryan Werner

Jonny Schneider

June 5, 2005

Abstract:

This paper explains the burr puzzle piece production system. Specifically it details and describes the process of our automation and how we accomplish the given criterion.

How do you specify cut pattern?

Introduction:

Awesome-o is a burr puzzle piece production system that meets specific criteria for entry and exit of blocks, part quality, dust collection, and safety.

Background Information:

Our strategy was inspired from two primary sources, the ideas of past students and from a memo containing five criteria. Those criteria were:

1. The ability to cut one, two, three, or four notches .375 inches square on a single face of each block. These cuts must create puzzles that maintain ninety-five percent assembly success rates.
2. The blocks are to travel in and out by belt conveyor (oriented in the long direction).
3. The machine must be able of either (we picked the latter):
    a. Accepting and operating on blocks that have already had two notches cut on one face OR
    b. Cut notches on two faces by rotating the block on its long axis within the machine.
4. Dust collection must be integrated in the machine.
5. Safety is a necessary consideration and as such jams must be minimized and clearing jams made safe and easy.

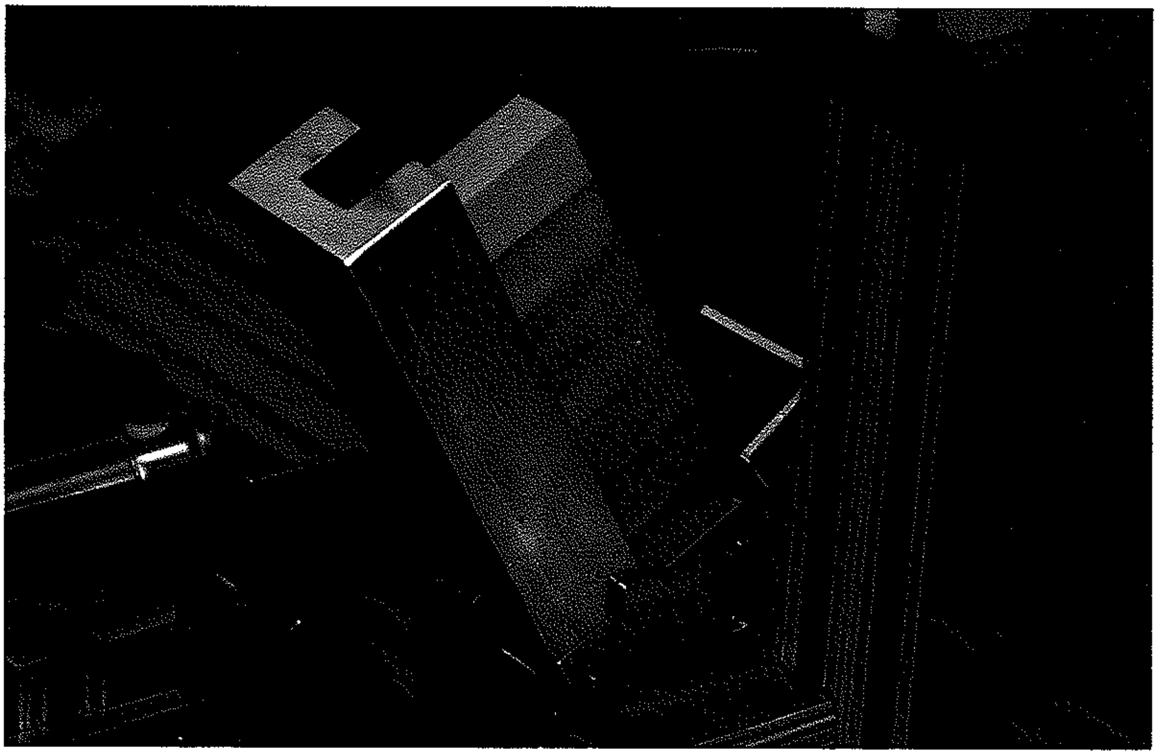
While the above criteria got us moving, it wasn't until we observed the projects from the previous year and compared its abilities to our needs that our strategy started to emerge.

Design and Process:

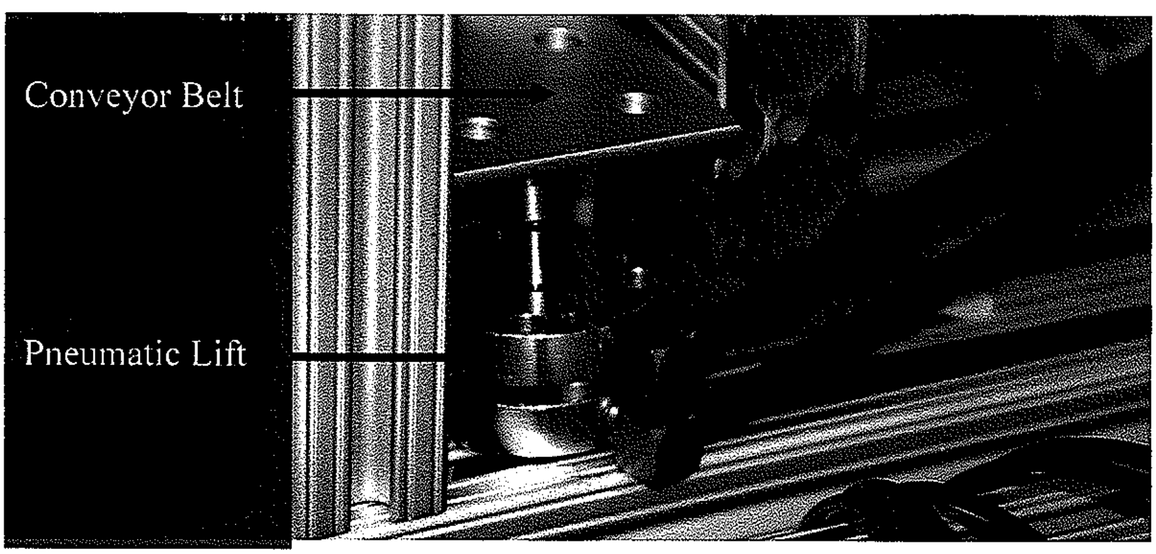
Although many of the ideas used the past year were helpful, the criteria motivated us in a new direction with the decision to accomplish a two-face cut procedure. This meant rotating the block within our machine. For a while we considered a design that would move the block from the belt to the router then rotate it using the lead screw, return it to the belt and then bring it to the router a second and final time for more cuts. This idea seemed inefficient by using unnecessary movement of the block to accomplish the rotation.

Our machine used the following process:

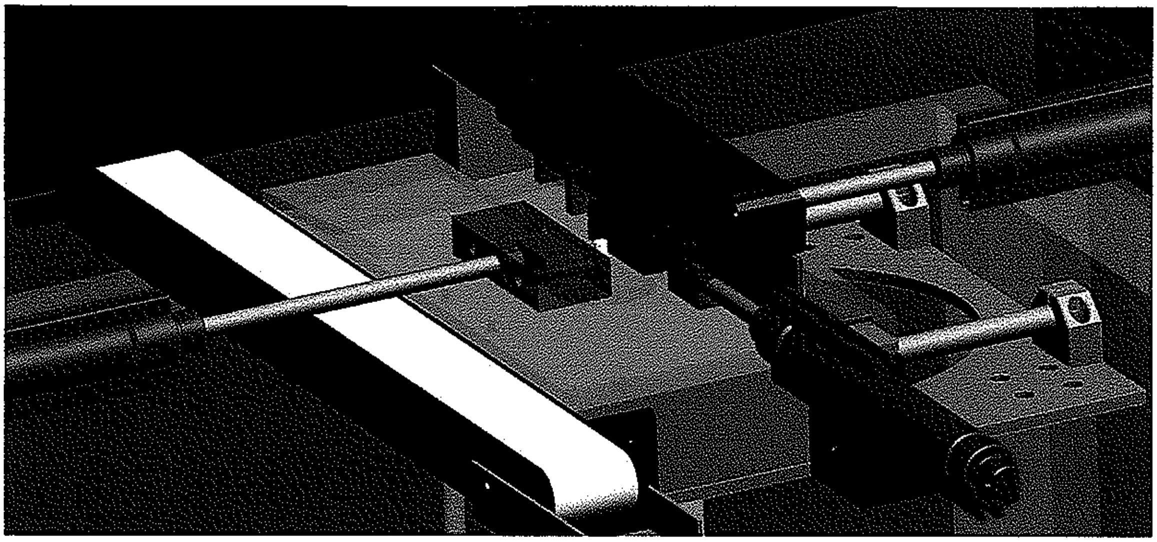
1. BLOCK LOADER and FLOW CONTROL
The blocks were placed in to the block loader, which combined with two pneumatic actuators, provided flow control at the beginning of the conveyor.



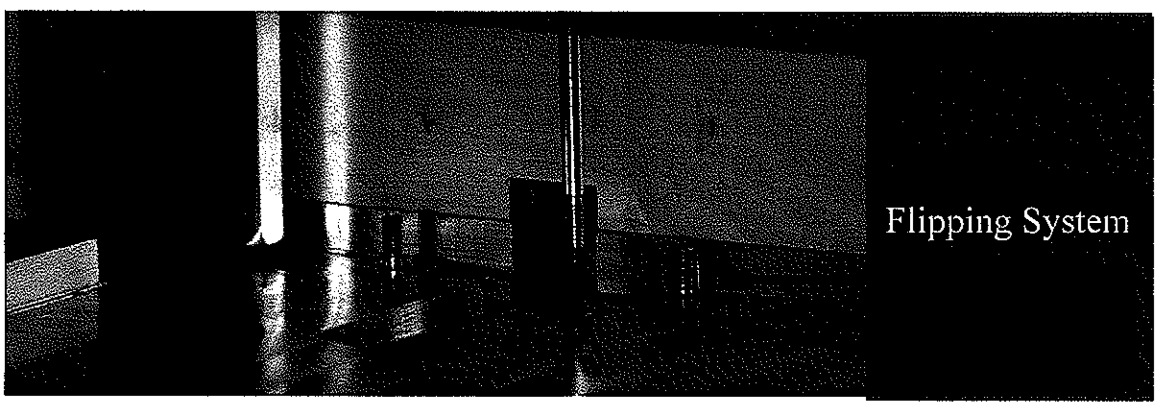
2. PIVOT GATE and CONVEYOR LIFT
Once through flow control, the block was stopped by the pivot gate while it waited to be pushed by the clamp over to the router. In order to push the block from the conveyor on to the cut plate easily, the conveyor was raised by a pneumatic lift.



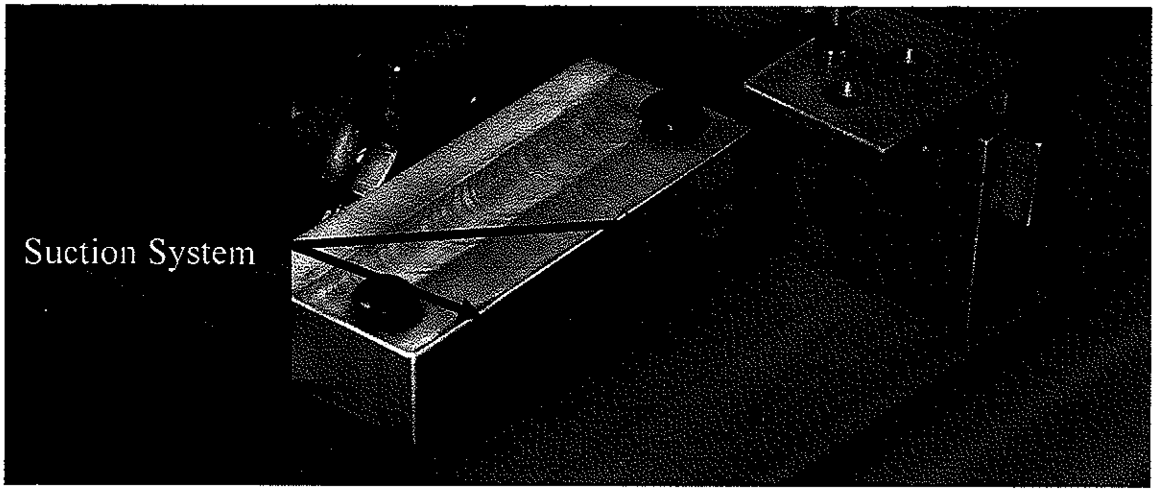
3. CLAMPING AND CUTTING
The block is now pushed over on the cut plate to prepare it for cutting. It is held in place by three pneumatics clamps and an electric lead screw.



4. FLIPPING SYSTEM
It is at this point that we hope to save time over our initial design by retracting the clamps enough to allow two flippers to lift the block from below, rolling it ninety degrees. With the block now rolled the clamps could quickly return and cutting could continue.



5. SUCTION SYSTEM and PIVOT GATE
With cutting finished, the block is ready to be returned to the belt. This is done using another new idea we incorporated in our machine. The same clamp that moved the block from the belt to the router was used to return the block to the belt. By adding two suction devices to the front of the clamp, the block could now be pulled back to the belt for its exit. With it back on the belt the pivot gate can be lifted to allow the block further down the conveyor.



6. BEHIND THE SCENES
   a. WIRING
   The actuators were wired so that an E-Stop switch could cut off power to most of the actuators without disturbing the PLC. The only exception to this is the router motor itself. In order to protect the PLC from carrying accidental high currents, all positive ("hot") wires were fused before connecting to the PLC.

   b. INPUTS
   Our project used ten inputs on the PLC. These inputs were used for four optical sensors, two induction sensors, and three Hall effect sensors on the cylinders. The last PLC input was a "Position Done" signal from the intelligent slider's controller. Even though only some of the sensors were sinking type, we wired the PLC so that it sources to all sensors.

   c. OUTPUTS
   We used all sixteen outputs on the PLC. These outputs controlled nine cylinders, two blower/sucker devices, and four intelligent slider commands. One of the cylinders used two air valves (and therefore two PLC outputs) in order to "float". Of the four wires controlling the intelligent slider, three were used for communicating desired position (three bits = eight positions), and one used for activating the slider. All the

relays and intelligent slider inputs were sourcing current, because the PLC required that it sink all of the outputs.

d. CUTTING PLATE AND BASE PLATE
An early element of our strategic design dealt with how we would assembly and fabricate things accurately. Knowing we had the Haas mill at our disposal we set to learning feature cam to create accurately dimensioned base plates and cutting plates. In doing so we created a more rigid framework for better tolerancing. Diagrams of the base and cutting plates can be found in the appendix.

Conclusions and Recommendations:

This project was useful for learning about manufacturing and automation strategy. This project also gave us hands on fabrication, programming and cad experience.

We learned of possible ways of improving our project all the way along and certainly at the end during demonstrations.

- Our clamping system could hold the block better as it was noticed that the block moved a little on the first cut.
- It was attributed to both the clamping and a dull router bit that there were chips still attached to the block after the cutting process. This was an unexpected problem that caused trouble for our flippers and our suction system. The chips would cause the block to roll 180 degrees instead of 90.

Acknowledgements:

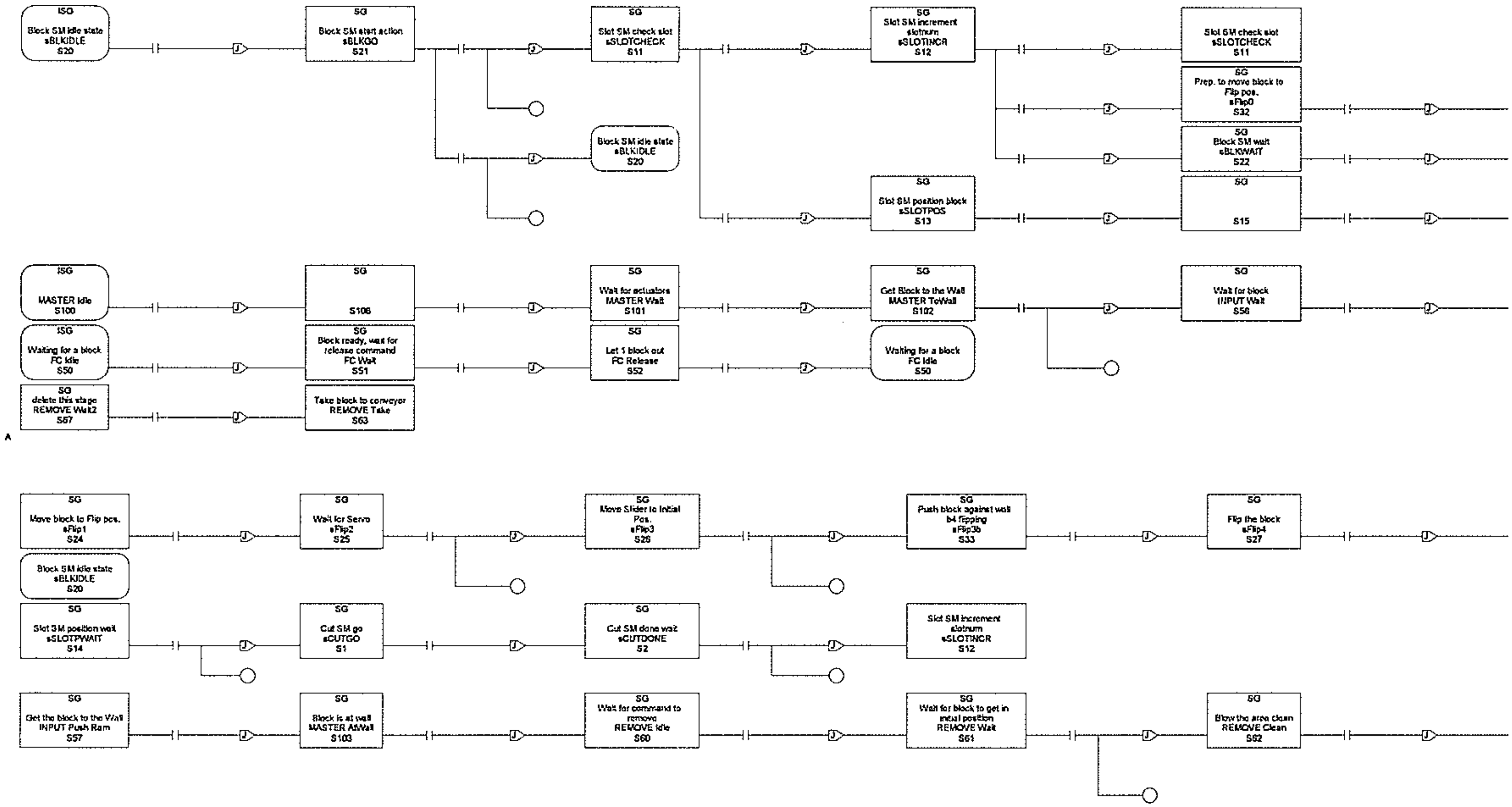
Our group would like to thank the Engineering department at Walla Walla College for providing a laboratory in which we could perform these experiments. We would also like to thank the people of TSS for supplies and equipment. We are grateful to Ralph Stirling for ideas and support. Finally, we would like to thank Mrs. Stirling for the sustenance that kept us inspired through the hard times.

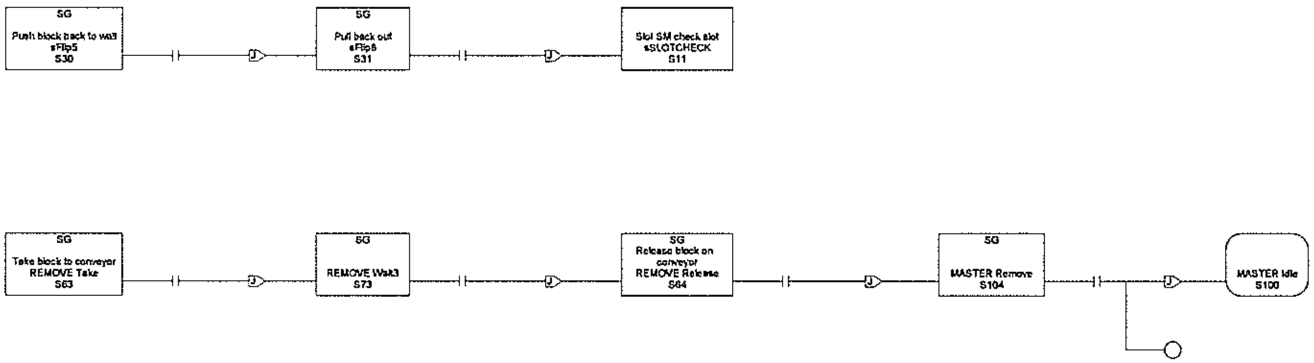
Description of Member Participation:

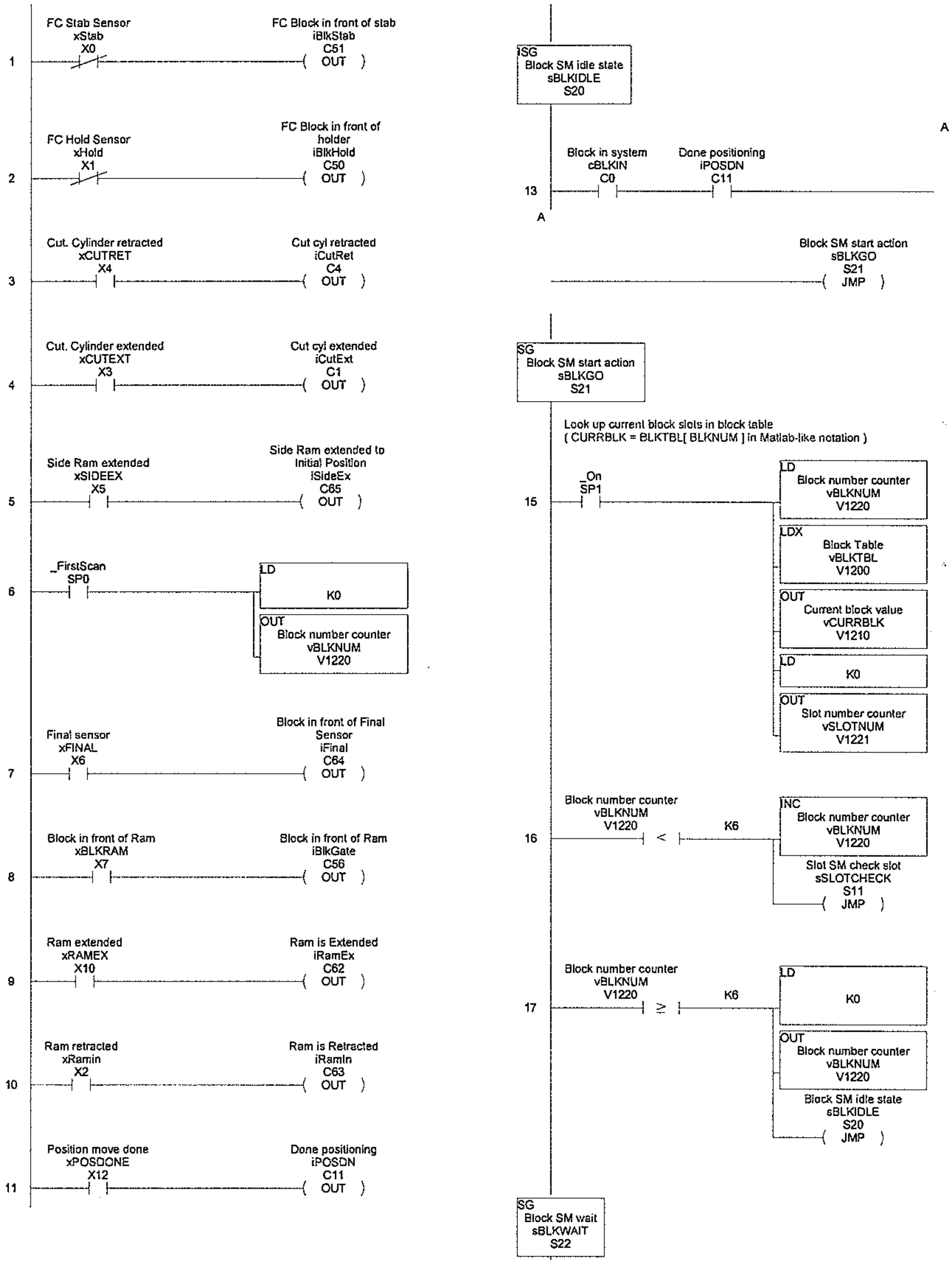
Matt Pawluk: Team Leader, ProE developer, Fabricator, testing, report write-up.
Kevin Cristman: Programming guru, wiring, sensors, testing.
Jon Schneider: idea development, report write-up, fabricator, testing.
Ryan Werner: fabricator, idea development, motivator, testing.

Appendix:

- State Diagram
- Ladder Logic
- Extra Diagrams


ISG Block SM idle state sBLKIDLE S20
SG Block SM start action sBLKGO S21
SG Slot SM check slot sSLOTCHECK S11
Block SM idle state sBLKIDLE S20
SG Slot SM increment slotnum sSLOTINCR S12
SG Slot SM position block sSLOTPOS S13
SG Slot SM check slot sSLOTCHECK S11
SG Prep. to move block to Flip pos. sFlip0 S32
SG Block SM wait sBLKWAIT S22
SG S15
ISG MASTER Idle S100
SG S106
SG Wait for actuators MASTER Wait S101
SG Get Block to the Wall MASTER ToWall S102
SG Wait for block INPUT Wait S56
ISG Waiting for a block FC Idle S50
SG Block ready, wait for release command FC Wait S51
SG Let 1 block out FC Release S52
Waiting for a block FC Idle S50
SG delete this stage REMOVE Wait2 S67
Take block to conveyor REMOVE Take S63
A
B
SG Move block to Flip pos. sFlip1 S24
SG Wait for Servo sFlip2 S25
SG Move Slider to Initial Pos. sFlip3 S26
SG Push block against wall b4 flipping sFlip3b S33
SG Flip the block sFlip4 S27
Block SM idle state sBLKIDLE S20
SG Slot SM position wait sSLOTPWAIT S14
SG Cut SM go sCUTGO S1
SG Cut SM done wait sCUTDONE S2
Slot SM increment slotnum sSLOTINCR S12
SG Get the block to the Wall INPUT Push Ram S57
SG Block is at wall MASTER AtWall S103
SG Wait for command to remove REMOVE Idle S60
SG Wait for block to get in initial position REMOVE Wait S61
SG Blow the area clean REMOVE Clean S62


SG
Push block back to wall
sFlip5
S30
SG
Pull back out
sFlip6
S31
Slot SM check slot
sSLOTCHECK
S11
SG
Take block to conveyor
REMOVE Take
S63
SG
REMOVE Wait3
S73
SG
Release block on conveyor
REMOVE Release
S64
SG
MASTER Remove
S104
MASTER Idle
S100


1
FC Stab Sensor
xStab
X0
FC Block in front of stab
iBlkStab
C51
OUT
2
FC Hold Sensor
xHold
X1
FC Block in front of holder
iBlkHold
C50
OUT
3
Cut. Cylinder retracted
xCUTRET
X4
Cut cyl retracted
iCutRet
C4
OUT
4
Cut. Cylinder extended
xCUTEXT
X3
Cut cyl extended
iCutExt
C1
OUT
5
Side Ram extended
xSIDEEX
X5
Side Ram extended to Initial Position
iSideEx
C65
OUT
6
_FirstScan
SP0
LD
K0
OUT
Block number counter
vBLKNUM
V1220
7
Final sensor
xFINAL
X6
Block in front of Final Sensor
iFinal
C64
OUT
8
Block in front of Ram
xBLKRAM
X7
Block in front of Ram
iBlkGate
C56
OUT
9
Ram extended
xRAMEX
X10
Ram is Extended
iRamEx
C62
OUT
10
Ram retracted
xRamin
X2
Ram is Retracted
iRamIn
C63
OUT
11
Position move done
xPOSDONE
X12
Done positioning
iPOSDN
C11
OUT
ISG
Block SM idle state
sBLKIDLE
S20
A
13
Block in system
cBLKIN
C0
Done positioning
iPOSDN
C11
A
Block SM start action
sBLKGO
S21
JMP
SG
Block SM start action
sBLKGO
S21
Look up current block slots in block table
(CURRBLK = BLKTBL[BLKNUM] in Matlab-like notation)
15
_On
SP1
LD
Block number counter
vBLKNUM
V1220
LDX
Block Table
vBLKTBL
V1200
OUT
Current block value
vCURRBLK
V1210
LD
K0
OUT
Slot number counter
vSLOTNUM
V1221
16
Block number counter
vBLKNUM
V1220
<
K6
INC
Block number counter
vBLKNUM
V1220
Slot SM check slot
sSLOTCHECK
S11
JMP
17
Block number counter
vBLKNUM
V1220
≥
K6
LD
K0
OUT
Block number counter
vBLKNUM
V1220
Block SM idle state
sBLKIDLE
S20
JMP
SG
Block SM wait
sBLKWAIT
S22

19 Block in system cBLKIN C0 (NC contact) — Block SM idle state sBLKIDLE S20 (JMP)

20 _On SP1 — Extend Side Ram oSide Y11 (RST)

SG
Slot SM check slot
sSLOTCHECK
S11

Get current slot value (1 or 0) from CURRBLK number. Cut slot if 1, skip if 0.

22 _On SP1 —
LD Current block value vCURRBLK V1210
SHFR Slot number counter vSLOTNUM V1221
OUT Current slot value vCURRSLOT V1211

23 Current slot bit (1=cut, 0=skip) bCURRSLOT B1211.0 (NC contact) — Slot SM increment slotnum sSLOTINCR S12 (JMP)

24 Current slot bit (1=cut, 0=skip) bCURRSLOT B1211.0 — Slot SM position block sSLOTPOS S13 (JMP)

A

25 Block number counter vBLKNUM V1220 = K1 — T0 (NC contact)

A

C100 (OUT)

26 _On SP1 — TMR T0 K5

SG
Slot SM increment slotnum
sSLOTINCR
S12

Increment slot number, or reset slot number if last slot

28 _On SP1 — INC Slot number counter vSLOTNUM V1221

A

29 Slot number counter vSLOTNUM V1221 ≠ K4

A

Slot number counter vSLOTNUM V1221 < K8 — Slot SM check slot sSLOTCHECK S11 (JMP)

30 Slot number counter vSLOTNUM V1221 = K4 — Prep. to move block to Flip pos. sFlip0 S32 (JMP)

31 Slot number counter vSLOTNUM V1221 = K8 — Block SM wait sBLKWAIT S22 (JMP)

SG
Slot SM position block
sSLOTPOS
S13

33 Slot number counter vSLOTNUM V1221 ≥ K4 —
LD Slot number counter vSLOTNUM V1221
SUBB K4
OUTD vTemp V1222

34 Slot number counter vSLOTNUM V1221 < K4 —
LD Slot number counter vSLOTNUM V1221
OUTD vTemp V1222

Start IAI slide positioning block for specified slot.

35 Ram is Extended iRamEx C62 (NC contact) —
LD vTemp V1222
OUTF Position bit0 oPOS0 Y14 K3
Start Servo move oPOSSTART Y17 (OUT)

36
_On SP1 —| |— Extend Ram oRamEx Y3 (RST)
Retract Ram oRamIn Y4 (OUT)
Extend Side Ram oSide Y11 (RST)

37
Done positioning iPOSDN C11 —|/|— Ram is Extended iRamEx C62 —|/|— A

A —— S15 (JMP)

SG S15

39
Done positioning iPOSDN C11 —|/|— Retract Ram oRamIn Y4 (OUT)

40
Done positioning iPOSDN C11 —| |— Extend Ram oRamEx Y3 (OUT)

41
Ram is Extended iRamEx C62 —| |— Done positioning iPOSDN C11 —| |— A

A —— Slot SM position wait sSLOTPWAIT S14 (JMP)

SG Slot SM position wait sSLOTPWAIT S14

43
Ram is Extended iRamEx C62 —| |— T5 —|/|— Retract Ram oRamIn Y4 (OUT)

Wait for IAI slide to finish motion.

44
Ram is Extended iRamEx C62 —|/|— Extend Side Ram oSide Y11 (SET)
TMR T5 K5

45
T5 —| |— Extend Ram oRamEx Y3 (SET)
Cut SM go sCUTGO S1 (JMP)

SG Cut SM go sCUTGO S1

Cut block.

47
Ram is Extended iRamEx C62 —| |— Cutting Cylinder oRoute Y5 (OUT)
TMR T12 K6

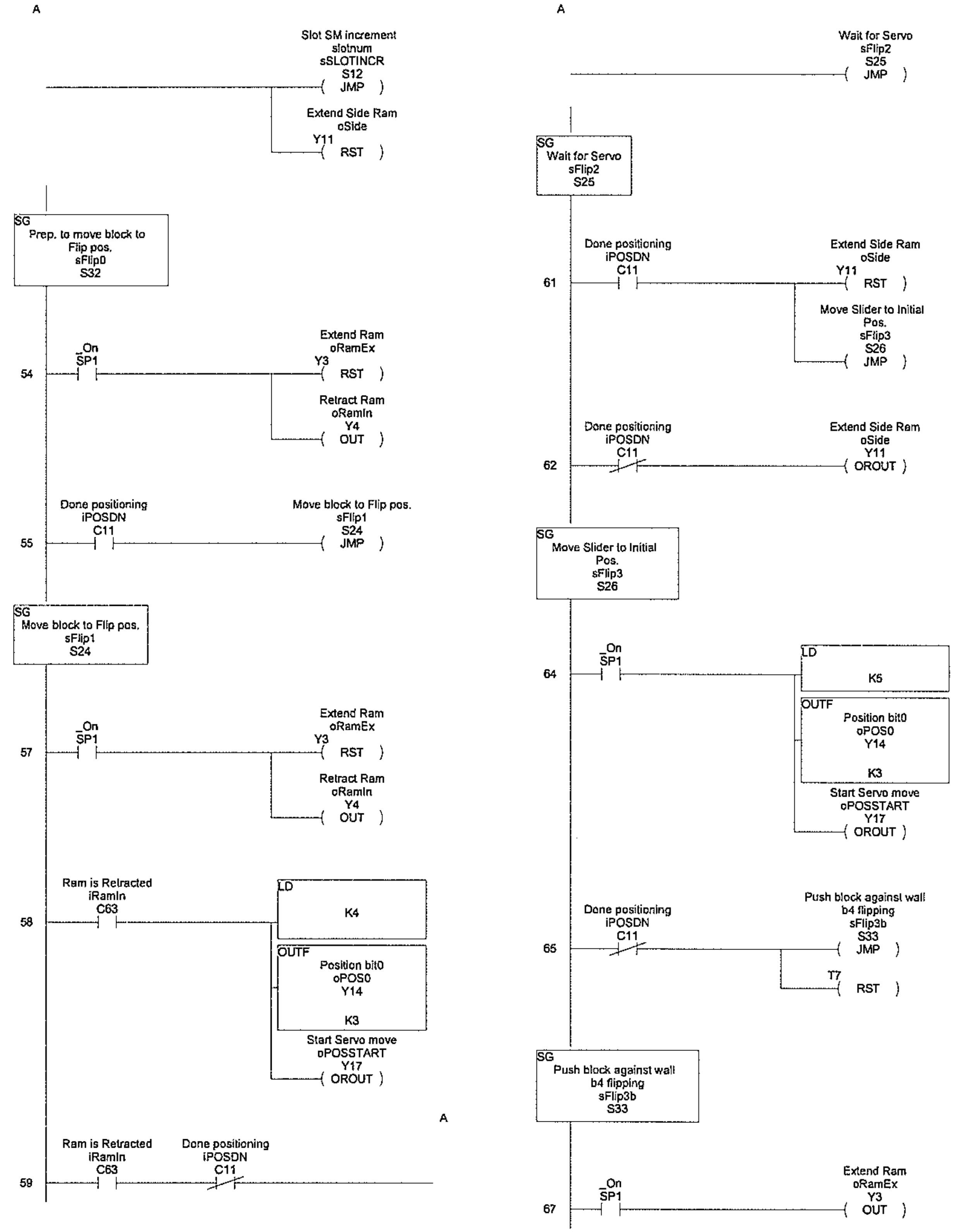
48
T12 —| |— Extend Top Clamp oTClamp Y0 (OUT)

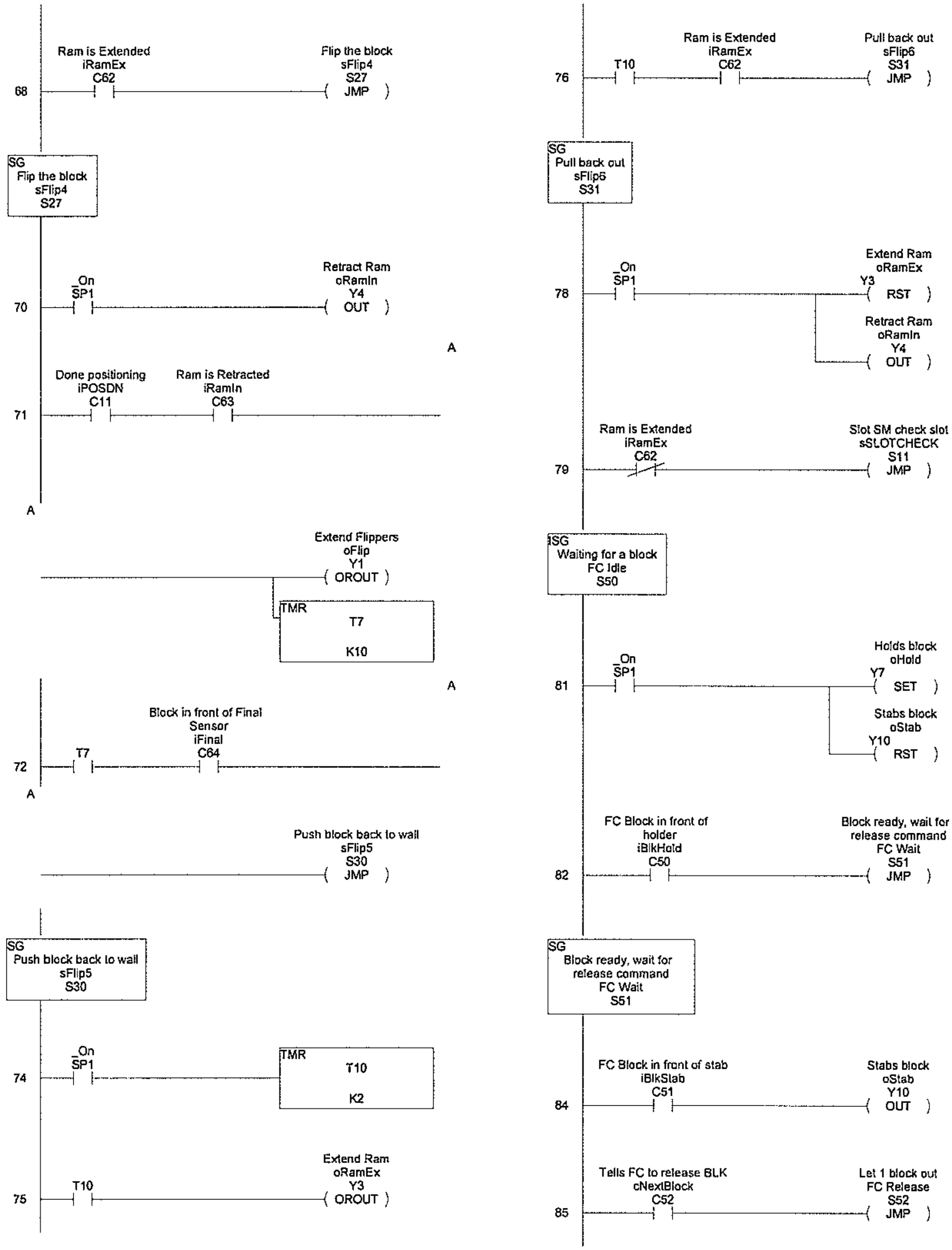
49
Cut cyl extended iCutExt C1 —| |— Cut SM done wait sCUTDONE S2 (JMP)

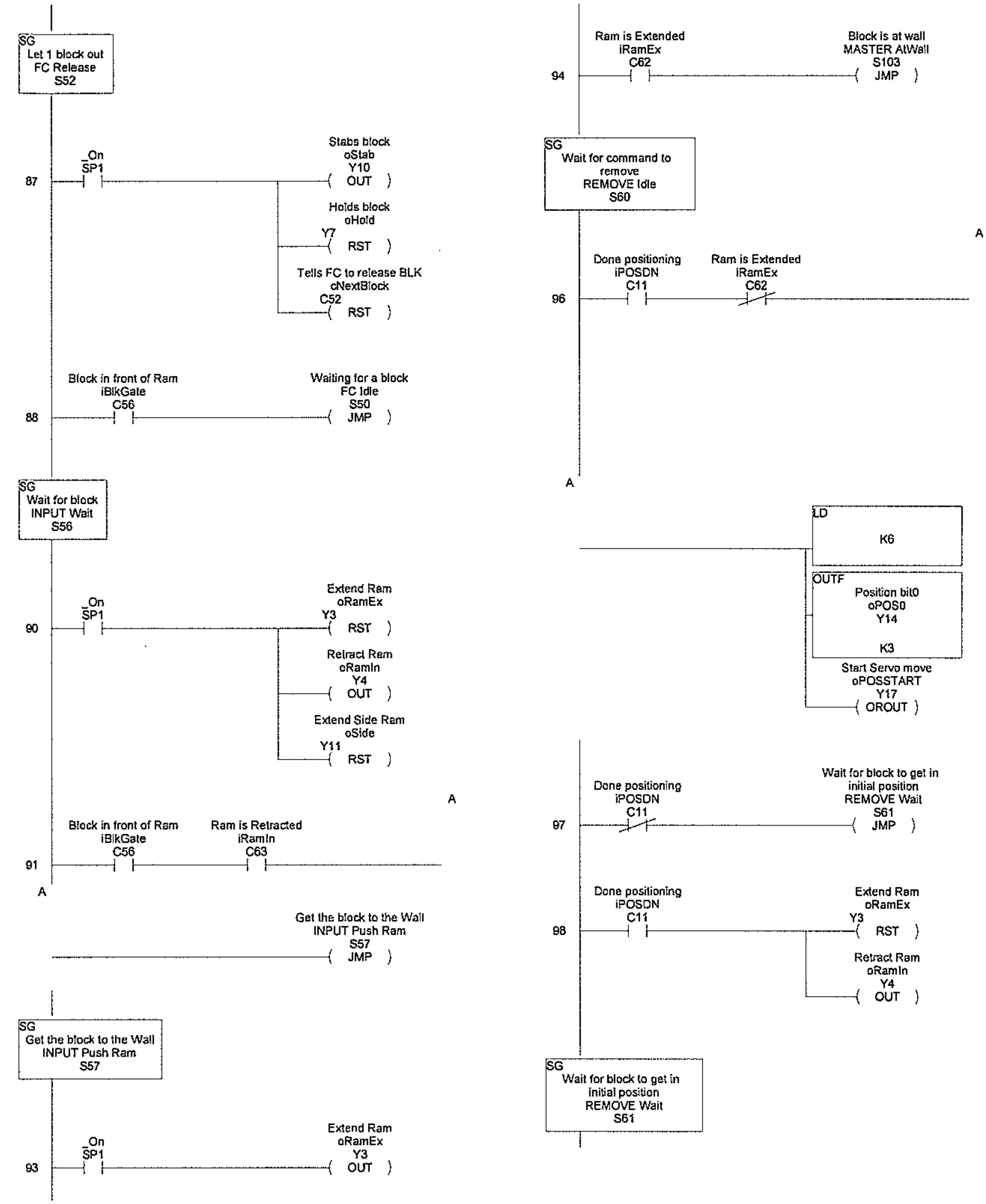
SG Cut SM done wait sCUTDONE S2

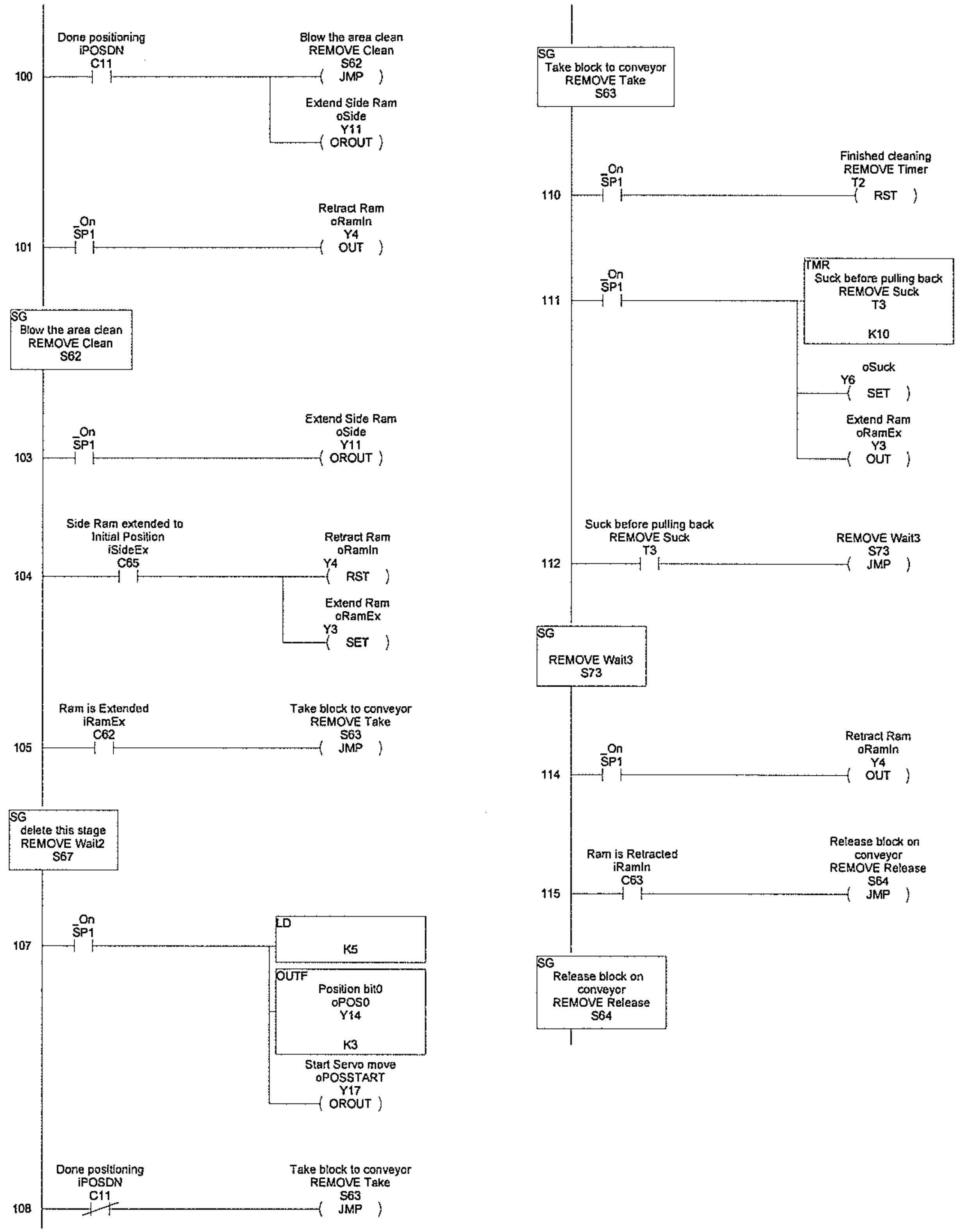
51
_On SP1 —| |— Extend Top Clamp oTClamp Y0 (OUT)

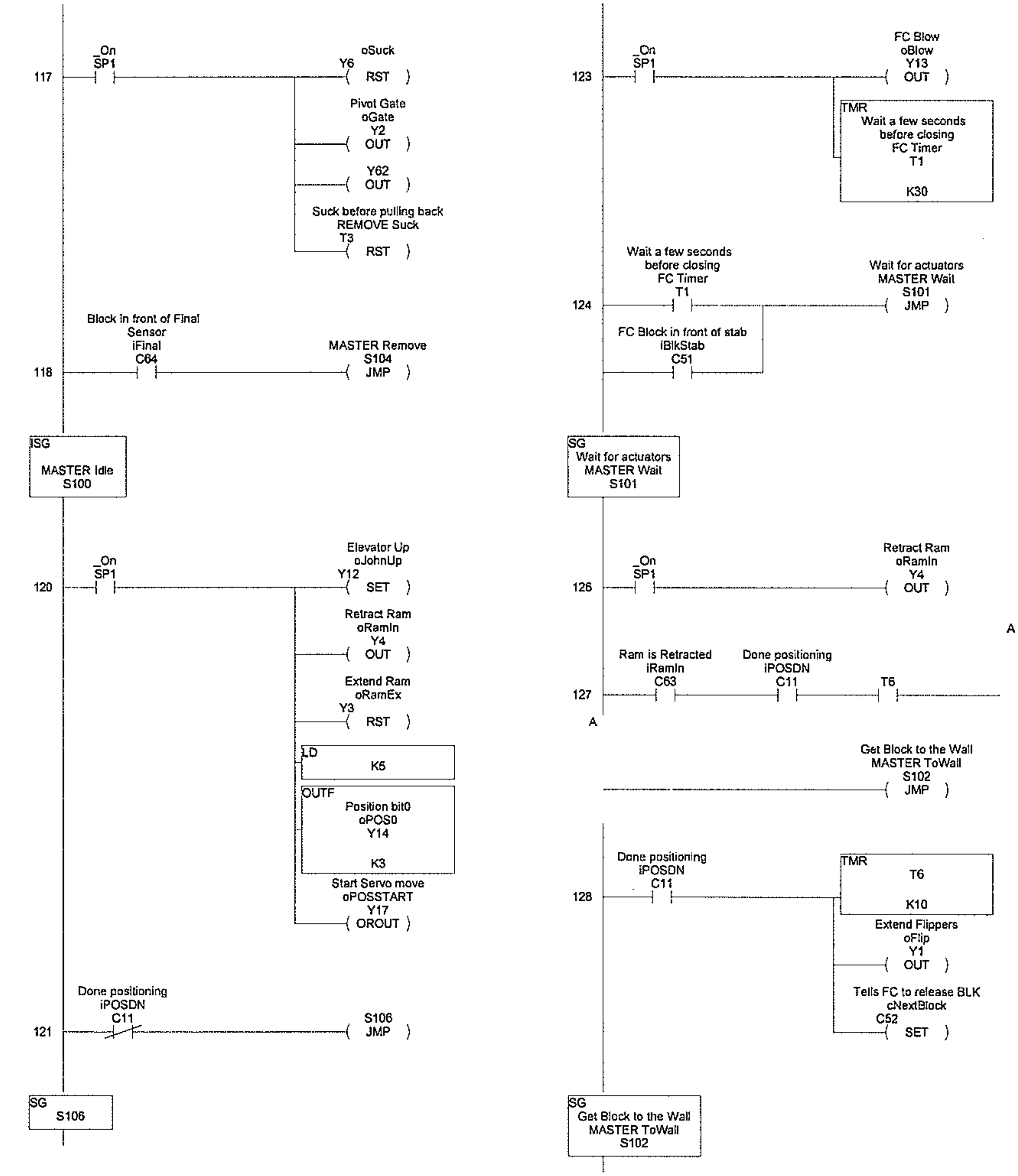
52
Cut cyl retracted iCutRet C4 —| |— Block in front of Final Sensor iFinal C64 —| |— A

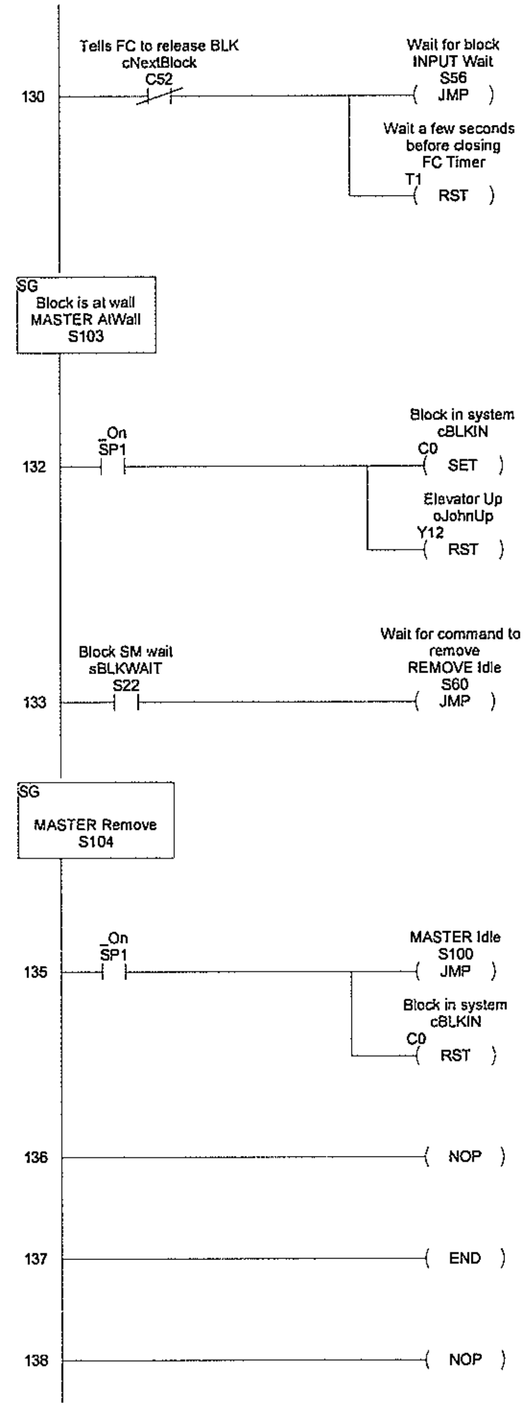

A
Slot SM increment
slotnum
sSLOTINCR
S12
(JMP)
Extend Side Ram
oSide
Y11
(RST)
SG
Prep. to move block to Flip pos.
sFlip0
S32
54
_On
SP1
Extend Ram
oRamEx
Y3
(RST)
Retract Ram
oRamIn
Y4
(OUT)
55
Done positioning
iPOSDN
C11
Move block to Flip pos.
sFlip1
S24
(JMP)
SG
Move block to Flip pos.
sFlip1
S24
57
_On
SP1
Extend Ram
oRamEx
Y3
(RST)
Retract Ram
oRamIn
Y4
(OUT)
58
Ram is Retracted
iRamIn
C63
LD
K4
OUTF
Position bit0
oPOS0
Y14
K3
Start Servo move
oPOSSTART
Y17
(OROUT)
A
59
Ram is Retracted
iRamIn
C63
Done positioning
iPOSDN
C11
A
Wait for Servo
sFlip2
S25
(JMP)
SG
Wait for Servo
sFlip2
S25
61
Done positioning
iPOSDN
C11
Extend Side Ram
oSide
Y11
(RST)
Move Slider to Initial Pos.
sFlip3
S26
(JMP)
62
Done positioning
iPOSDN
C11
Extend Side Ram
oSide
Y11
(OROUT)
SG
Move Slider to Initial Pos.
sFlip3
S26
64
_On
SP1
LD
K5
OUTF
Position bit0
oPOS0
Y14
K3
Start Servo move
oPOSSTART
Y17
(OROUT)
65
Done positioning
iPOSDN
C11
Push block against wall b4 flipping
sFlip3b
S33
(JMP)
T7
(RST)
SG
Push block against wall b4 flipping
sFlip3b
S33
67
_On
SP1
Extend Ram
oRamEx
Y3
(OUT)

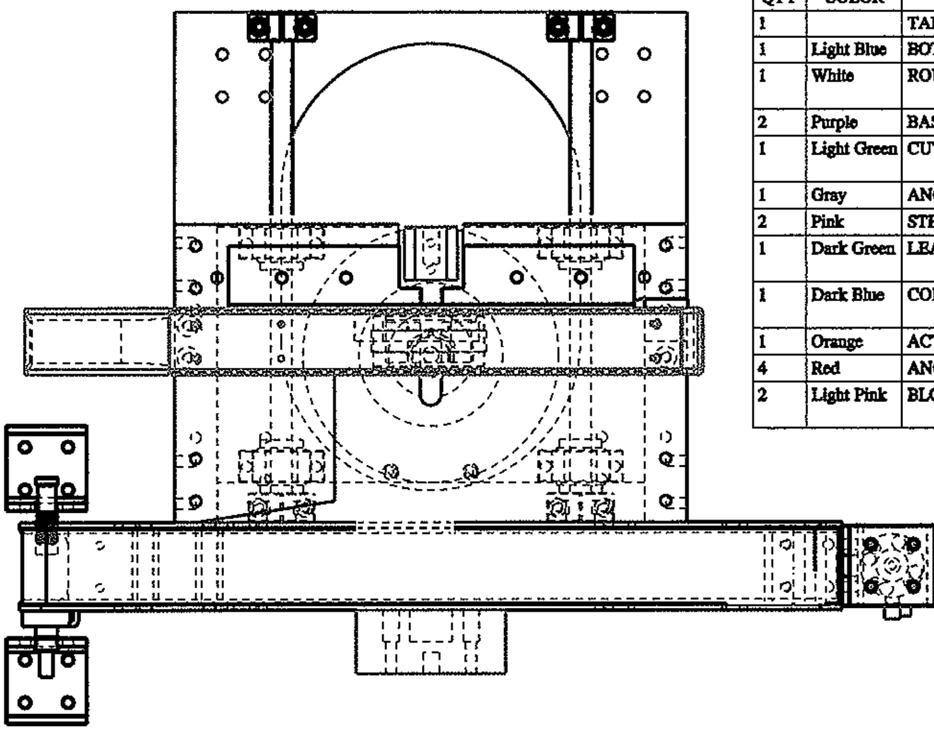

68
Ram is Extended
iRamEx
C62
Flip the block
sFlip4
S27
JMP
SG
Flip the block
sFlip4
S27
70
_On
SP1
Retract Ram
oRamIn
Y4
OUT
A
71
Done positioning
iPOSDN
C11
Ram is Retracted
iRamIn
C63
A
Extend Flippers
oFlip
Y1
OROUT
TMR
T7
K10
A
72
T7
Block in front of Final
Sensor
iFinal
C64
A
Push block back to wall
sFlip5
S30
JMP
SG
Push block back to wall
sFlip5
S30
74
_On
SP1
TMR
T10
K2
75
T10
Extend Ram
oRamEx
Y3
OROUT
76
T10
Ram is Extended
iRamEx
C62
Pull back out
sFlip6
S31
JMP
SG
Pull back out
sFlip6
S31
78
_On
SP1
Extend Ram
oRamEx
Y3
RST
Retract Ram
oRamIn
Y4
OUT
79
Ram is Extended
iRamEx
C62
Slot SM check slot
sSLOTCHECK
S11
JMP
SG
Waiting for a block
FC Idle
S50
81
_On
SP1
Holds block
oHold
Y7
SET
Stabs block
oStab
Y10
RST
82
FC Block in front of
holder
iBlkHold
C50
Block ready, wait for
release command
FC Wait
S51
JMP
SG
Block ready, wait for
release command
FC Wait
S51
84
FC Block in front of stab
iBlkStab
C51
Stabs block
oStab
Y10
OUT
85
Tells FC to release BLK
cNextBlock
C52
Let 1 block out
FC Release
S52
JMP


SG
Let 1 block out
FC Release
S52
87
_On
SP1
Stabs block
oStab
Y10
OUT
Holds block
oHold
Y7
RST
Tells FC to release BLK
cNextBlock
C52
RST
88
Block in front of Ram
iBlkGate
C56
Waiting for a block
FC Idle
S50
JMP
SG
Wait for block
INPUT Wait
S56
90
_On
SP1
Extend Ram
oRamEx
Y3
RST
Retract Ram
oRamIn
Y4
OUT
Extend Side Ram
oSide
Y11
RST
A
91
Block in front of Ram
iBlkGate
C56
Ram is Retracted
iRamIn
C63
A
Get the block to the Wall
INPUT Push Ram
S57
JMP
SG
Get the block to the Wall
INPUT Push Ram
S57
93
_On
SP1
Extend Ram
oRamEx
Y3
OUT
94
Ram is Extended
iRamEx
C62
Block is at wall
MASTER AtWall
S103
JMP
SG
Wait for command to
remove
REMOVE Idle
S60
A
96
Done positioning
iPOSDN
C11
Ram is Extended
iRamEx
C62
A
LD
K6
OUTF
Position bit0
oPOS0
Y14
K3
Start Servo move
oPOSSTART
Y17
OROUT
97
Done positioning
iPOSDN
C11
Wait for block to get in
initial position
REMOVE Wait
S61
JMP
98
Done positioning
iPOSDN
C11
Extend Ram
oRamEx
Y3
RST
Retract Ram
oRamIn
Y4
OUT
SG
Wait for block to get in
initial position
REMOVE Wait
S61

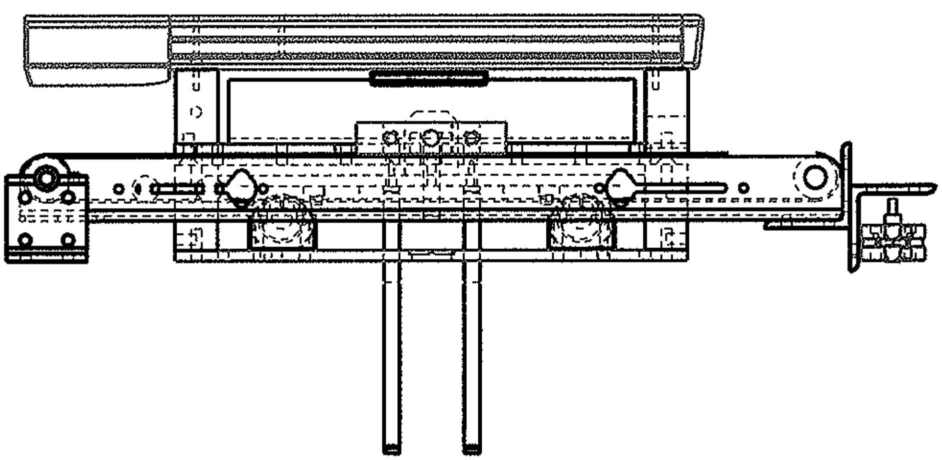

100
Done positioning
iPOSDN
C11
Blow the area clean
REMOVE Clean
S62
JMP
Extend Side Ram
oSide
Y11
OROUT
101
_On
SP1
Retract Ram
oRamIn
Y4
OUT
SG
Blow the area clean
REMOVE Clean
S62
103
_On
SP1
Extend Side Ram
oSide
Y11
OROUT
104
Side Ram extended to
Initial Position
iSideEx
C65
Retract Ram
oRamIn
Y4
RST
Extend Ram
oRamEx
Y3
SET
105
Ram is Extended
iRamEx
C62
Take block to conveyor
REMOVE Take
S63
JMP
SG
delete this stage
REMOVE Wait2
S67
107
_On
SP1
LD
K5
OUTF
Position bit0
oPOS0
Y14
K3
Start Servo move
oPOSSTART
Y17
OROUT
108
Done positioning
iPOSDN
C11
Take block to conveyor
REMOVE Take
S63
JMP
SG
Take block to conveyor
REMOVE Take
S63
110
_On
SP1
Finished cleaning
REMOVE Timer
T2
RST
111
_On
SP1
TMR
Suck before pulling back
REMOVE Suck
T3
K10
oSuck
Y6
SET
Extend Ram
oRamEx
Y3
OUT
112
Suck before pulling back
REMOVE Suck
T3
REMOVE Wait3
S73
JMP
SG
REMOVE Wait3
S73
114
_On
SP1
Retract Ram
oRamIn
Y4
OUT
115
Ram is Retracted
iRamIn
C63
Release block on
conveyor
REMOVE Release
S64
JMP
SG
Release block on
conveyor
REMOVE Release
S64

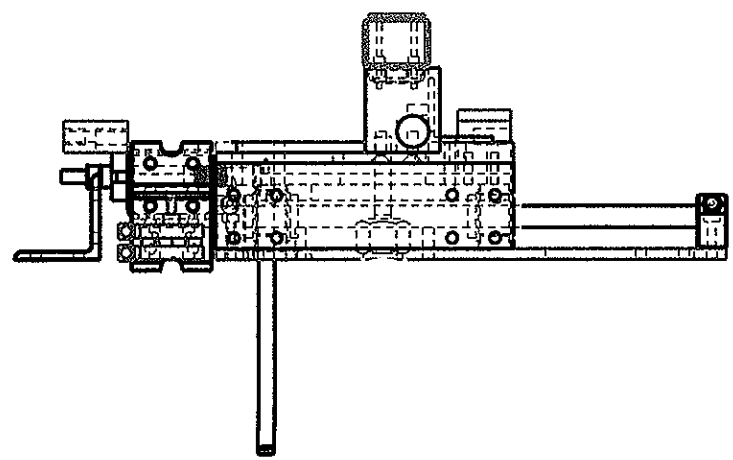

117
_On
SP1
oSuck
Y6
RST
Pivot Gate
oGate
Y2
OUT
Y62
OUT
Suck before pulling back
REMOVE Suck
T3
RST
118
Block in front of Final
Sensor
iFinal
C64
MASTER Remove
S104
JMP
ISG
MASTER Idle
S100
120
_On
SP1
Elevator Up
oJohnUp
Y12
SET
Retract Ram
oRamIn
Y4
OUT
Extend Ram
oRamEx
Y3
RST
LD
K5
OUTF
Position bit0
oPOS0
Y14
K3
Start Servo move
oPOSSTART
Y17
OROUT
121
Done positioning
iPOSDN
C11
S106
JMP
SG
S106
123
_On
SP1
FC Blow
oBlow
Y13
OUT
TMR
Wait a few seconds
before closing
FC Timer
T1
K30
124
Wait a few seconds
before closing
FC Timer
T1
FC Block in front of stab
iBlkStab
C51
Wait for actuators
MASTER Wait
S101
JMP
SG
Wait for actuators
MASTER Wait
S101
126
_On
SP1
Retract Ram
oRamIn
Y4
OUT
A
127
Ram is Retracted
iRamIn
C63
Done positioning
iPOSDN
C11
T6
A
Get Block to the Wall
MASTER ToWall
S102
JMP
128
Done positioning
iPOSDN
C11
TMR
T6
K10
Extend Flippers
oFlip
Y1
OUT
Tells FC to release BLK
cNextBlock
C52
SET
SG
Get Block to the Wall
MASTER ToWall
S102

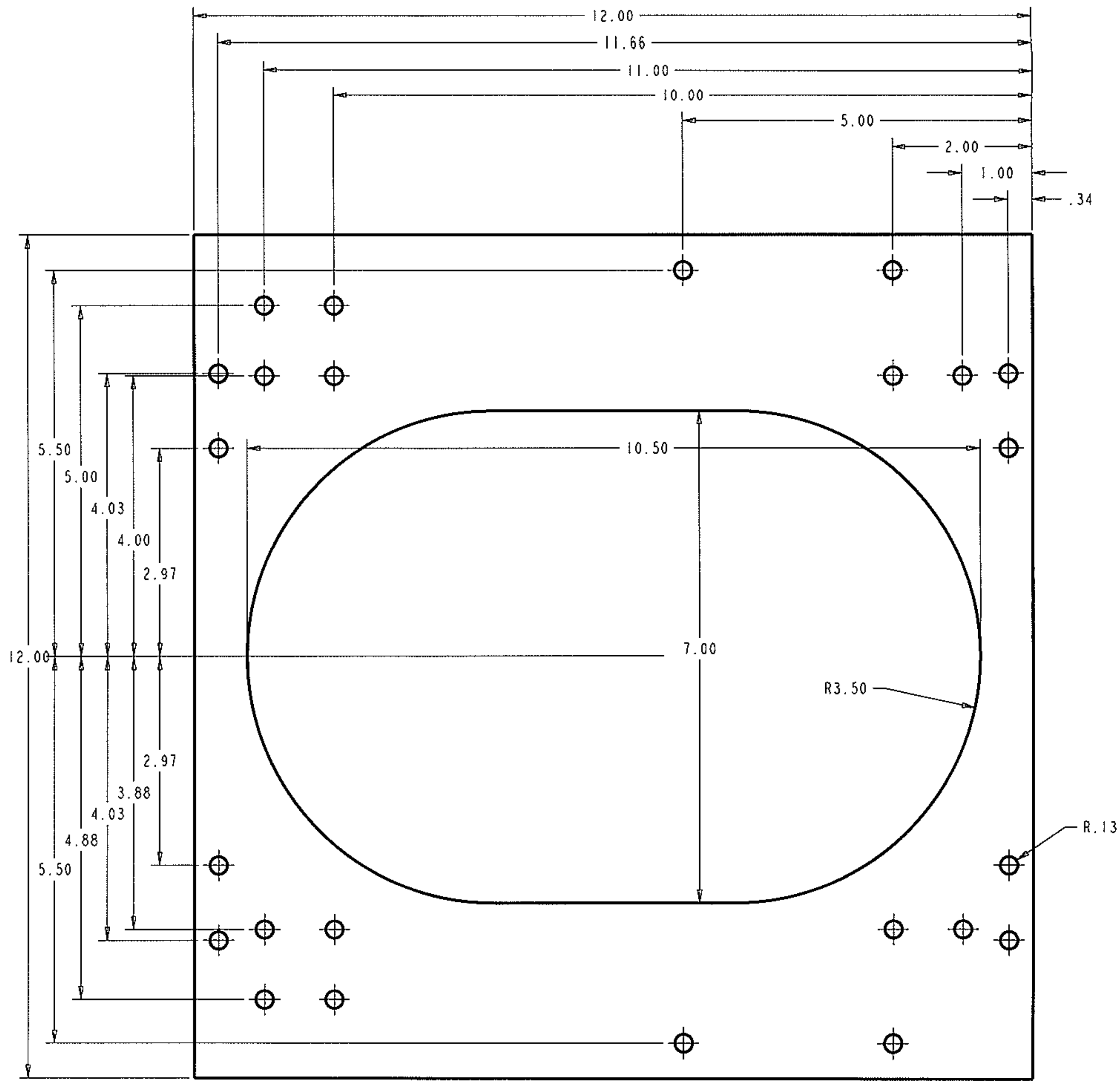

130
Tells FC to release BLK
cNextBlock
C52
Wait for block
INPUT Wait
S56
JMP
Wait a few seconds
before closing
FC Timer
T1
RST
SG
Block is at wall
MASTER AtWall
S103
132
_On
SP1
Block in system
cBLKIN
C0
SET
Elevator Up
oJohnUp
Y12
RST
133
Block SM wait
sBLKWAIT
S22
Wait for command to
remove
REMOVE Idle
S60
JMP
SG
MASTER Remove
S104
135
_On
SP1
MASTER Idle
S100
JMP
Block in system
cBLKIN
C0
RST
136
NOP
137
END
138
NOP

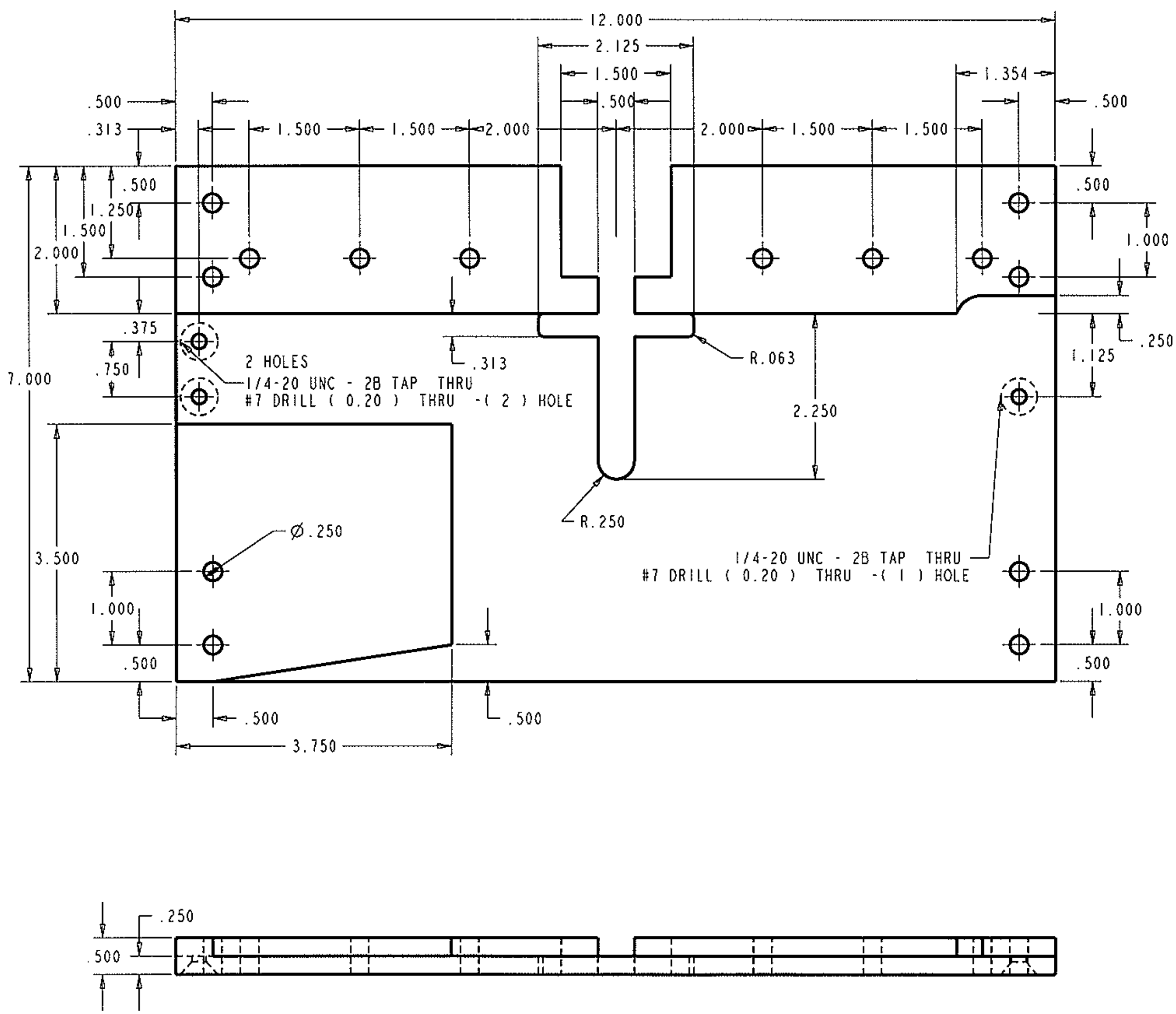


QTY	COLOR	PART NAME	DESCRIPTION
1		TABLE	Provides support for all parts and sensors.
1	Light Blue	BOTTOM_PLATE	Medium betwen ROUTER, CUT_PLATE and TABLE.
1	White	ROUTER	Router for Block cutting. Base for ACTUATOR_FLIPPER.
2	Purple	BASE_STEP	Support for the CUT_PLATE.
1	Light Green	CUT_PLATE	Part on which block cutting occurs. Base for STEPPER_MOTOR.
1	Gray	ANGLE_BAR	Side suport for Block during cutting.
2	Pink	STEPPER_MOTOR_MOUNT[2]	Support and spacing from CUT_PLATE and STEPPER_MOTOR.
1	Dark Green	LEAD_SCREW_MOTOR	Includes STEPPER_MOTOR, STEPPER_MOTOR_TAB, and LEAD_SCREW_PUSHTAB. Moves the Block axially for prescion cutting.
1	Dark Blue	CONVEYOUR_SMALL	Includes CONVEYOR_SMALL and CONVEYOR_SCREW. Moves the Block from initial stage to complete stage.
1	Orange	ACTUATOR_PANCAKE	Pushes the CONVEYOR_SMALL up and down for smooth Block positioning.
4	Red	ANGLE_4HOLE_BRACE	Provides various support for CONVEYOUR_SMALL.
2	Light Pink	BLOCK_CLAMP	Includes BLOCK_SIDECLAMP and BLOCK_ENDCLAMP. Pushes Block into cutting area.






12.00
11.66
11.00
10.00
5.00
2.00
1.00
.34
5.50
5.00
4.03
4.00
2.97
10.50
7.00
12.00
R3.50
2.97
3.88
4.03
4.88
5.50
R.13


12.000
2.125
1.500
.500
1.354
.313
1.500
2.000
1.250
.375
.750
7.000
2 HOLES
1/4-20 UNC - 2B TAP THRU
#7 DRILL (0.20) THRU -(2) HOLE
.313
R.063
2.250
R.250
Ø.250
3.500
1.000
1/4-20 UNC - 2B TAP THRU
#7 DRILL (0.20) THRU -(1) HOLE
1.125
.250
3.750