SimpleSip Project

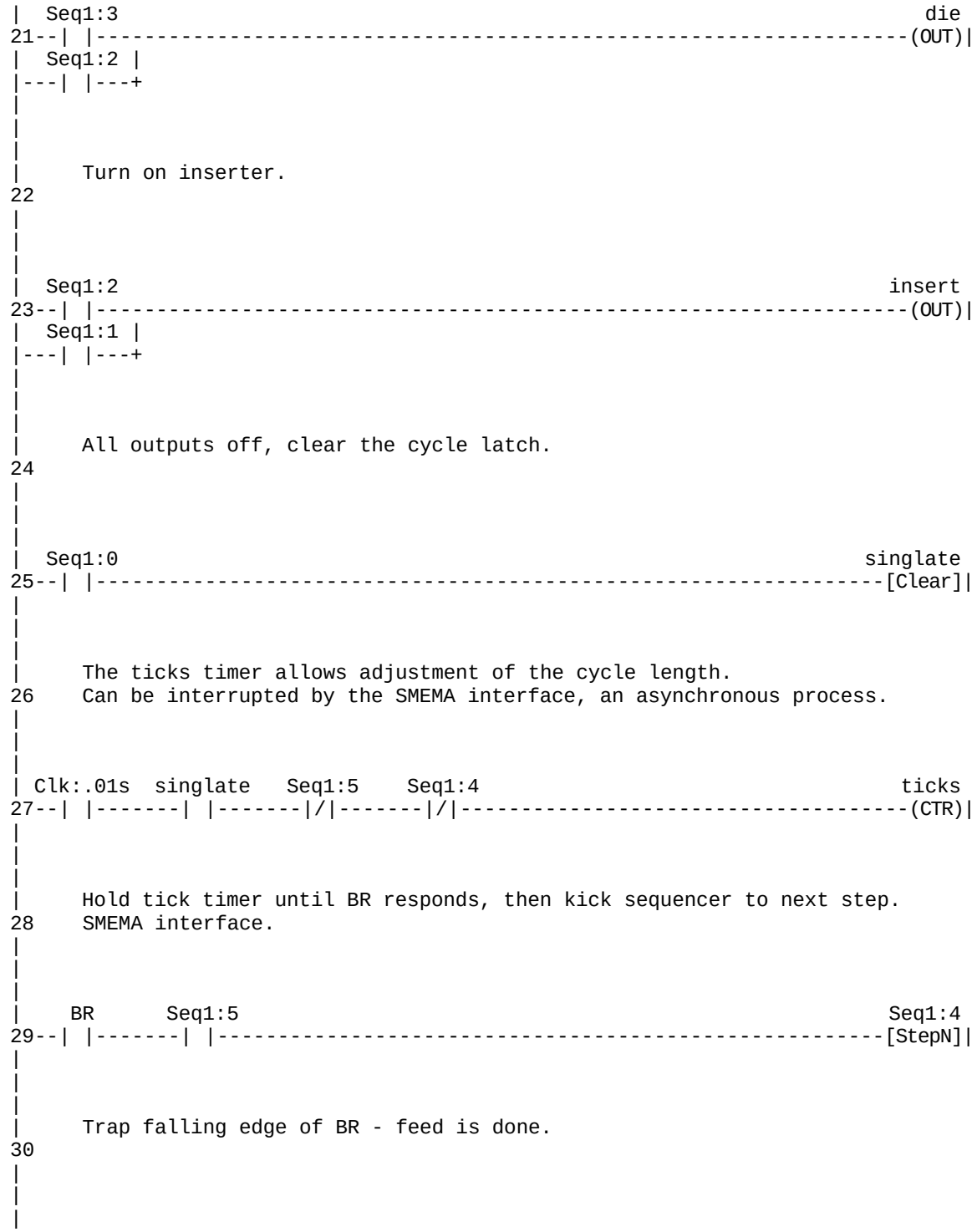
Michael Dennis

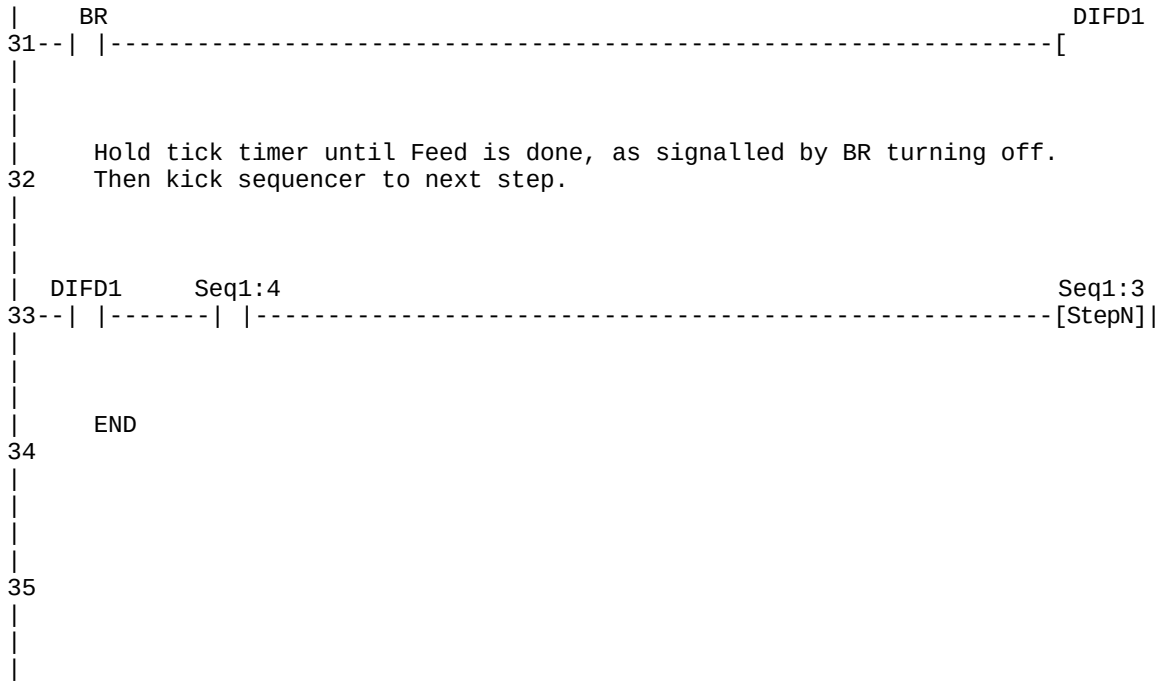
Die-Tech, Inc.
295 Sipe Road
York Haven, PA 17601

See us on the web at www.die-tech.com
info@die-tech.com

```
|     SimpleSIP - copyright 1999 Die-Tech, Inc.
1     m. dennis programmer
|     A minimalist SIP machine, which uses the E10 as a fixed sequencer.
|
|
|     This file may be freely copied and distributed as long as the
2     copyright and this notice are left intact. If you use the code in a
|     project, please tell me about it at info@die-tech.com.
|
|
|     Any use is at your risk - always provide adequate safeguards when
3     developing software that runs machinery.
|
|
|
|     ------------------------------------------------------------------------
4     rev 8 May 99 - mdd - beta
|
|
|
|     rev 12 May 99 - mdd - extended with count.pe3, a single-pin feed
5     routine cascaded via MA signal and BR input. A second E10
|     runs the feed cycle.
|
|
|     rev 14 May 99 - mdd - Recoded to conform to SMEMA Spec.
6
|
|
|
|     Assumptions:
7     Tooling is adjustable for chip placement.
|     Cutoff tool travels with die. Feed is escapement, 1 pin per pulse.
|
|
|     Operator places chip and removes it.
8     Multiple chips may be placed if the die/leadframe combination allows.
|     Estop cuts power (electrical & air) to entire machine via relay.
|
|
|     Cycle:
9     1. Feed
|     2. Wait for BR
|
|
|     3. Clamp & Cut
10    4. Inserter in
|     5. Die off
|
|
|     6. Inserter out.
11
|
|
|
```

```
|     ------------------------------------------------------------------------
12    Run button is a foot pedal. Scan it once, so operator has to
|     release it between cycles. Ignore it during cycle.
|
|
|   run     singlate                                                  run1shot
13--| |-------|/|-----------------------------------------------------[
|
|
|
|     Latch for one machine cycle - this is a semi-auto machine.
14    While we're at it, point the sequencer to its first step.
|
|
|
| run1shot                                                            singlate
15--| |-------------------------------------------------------------[Latch]|
|                                                                   |  Seq1:5
|                                                                   +-[StepN]|
|
|
|
|     While cycling, ticks counts secs/100 and increments the seq1 count.
16    At each seq1, reset ticks so it can count down again.
|     Singulate interrupts the cycle at the last seq1 of the cycle.
|
|
|  ticks                                                                seq1
17--| |-------------------------------------------------------------(CTR)|
|                                                                   |   ticks
|                                                                   +-[RSctr]|
|
|
|
|     seq1 counter decrements from preset, so seq1 5 is first step.
18    Feed (MA) is turned on as needed.
|     Other PLC drives feed.
|
|
|  Seq1:5                                                               MA
19--| |-------------------------------------------------------------(OUT)|
|  Seq1:4 |
|---| |---+
|
|
|
|     Die is turned on and off as needed.
20
|
|
|
```


Seq1:3
die
(OUT)
Seq1:2
Turn on inserter.
Seq1:2
insert
(OUT)
Seq1:1
All outputs off, clear the cycle latch.
Seq1:0
singlate
[Clear]
The ticks timer allows adjustment of the cycle length.
Can be interrupted by the SMEMA interface, an asynchronous process.
Clk:.01s
singlate
Seq1:5
Seq1:4
ticks
(CTR)
Hold tick timer until BR responds, then kick sequencer to next step.
SMEMA interface.
BR
Seq1:5
Seq1:4
[StepN]
Trap falling edge of BR - feed is done.


BR
DIFD1
Hold tick timer until Feed is done, as signalled by BR turning off.
Then kick sequencer to next step.
DIFD1
Seq1:4
Seq1:3
[StepN]
END

Combined I/O Definition Table

I/O#	Input	Output	Relay	Timer	Value	Counter	Value
1	run	MA	reset_ct			seq1	5
2		die	singlate			ticks	50
3		insert	run1shot				
4			DIFD1				
5							
6	BR						
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							

```
|     COUNT.PE3
1     Copyright 1999 Die-Tech, Inc. m. dennis, programmer
|     The pulse output half of our Simple SIP machine.
|
|
|     ---------------------------------------------------------------------
2     This file may be freely copied and distributed as long as the
|     copyright and this notice are left intact.
|
|
|     You may use the code in your own projects, and if you do, please
3     tell me about it at info@die-tech.com. If you make modifications,
|     generate a new file and document the changes.
|
|
|     Any use of this code is at your own risk. Design adequate safeguards
4     into any software or device that controls machinery.
|     ---------------------------------------------------------------------
|
|
|     Needs another E10 to finsh the job. But the interface is
5     SMEMA compliant, so you could use the module anywhere you need
|     set of slow pulses.
|
|
|     The E10 does not allow you to change the set value of its counters,
6     but it does allow you to change the present value of Seq1 via
|     thumbwheel or other BCD input.
|
|
|     This code exploits that feature of Seq1 to give the operator a 0-199
7     pulse output. It does this by multiplexing two thumbwheels,
|     one for 1's and one for 10's. Without multiplexing, the input
|
|
|     would be limited to 15 counts. (0-15 represents the range of 4-bit
8     BCD). A third wheel has its 1's digit wired to input 6 to signal
|     100 pulses as needed.
|
|
|     revision history:
9     23 May 99 - mdd - first bug-free run after two weeks beta.
|     23 May 99 - mdd - added 3rd digit.
|
|
|     24 May 99 - mdd - eliminated short pulses and miscounts.
10
|
|
|
|     This rung patches a couple of characteristics of the E10:
11    1. You can't scan a thumbwheel in one scan, the inputs are too slow.
|     2. DIFU is not reliable when you're doing this much work.
|
|
```

```
|     MA is a SMEMA request for service. Our PLC ought to respond with
12    BR, and when it is finshed its cycle, clear BR. The the other
|     machine may proceed, and will drop MA.
|
|
|     The timer acts like a long DIFU, making sure that we get to scan the
13    thumbwheel before starting the pulse count.
|
|
|
|     Meanwhile, turn on the common line to the 1's thumbwheel.
14    Leave it on until we're ready to read 10's.
|
|
|
|    MA      sel10     Sel100                                              sel1
15--| |-------|/|-------|/|-------------------------------------------------(OUT)|
|                                                                        | OneShot
|                                                                        +---(TIM)|
|
|
|
|     This rung is funtionally the same as the previous one, but it
16    executes after we have delt with the 1's count.
|
|
|
|   exp1     Sel100      BR                                                sel10
17--| |-------|/|-------| |-------------------------------------------------(OUT)|
|  sel10  |                                                              | TenShot
|---| |---+                                                              +---(TIM)|
|
|
|
|     This rung executes after we have dealt with the 10's count.
18
|
|
|
|   exp2       BR                                                          Sel100
19--| |-------| |-----------------------------------------------------------(RLY)|
|  Sel100 |                                                              | 100Shot
|---| |---+                                                              +---(TIM)|
|
|
|
|     For the amount of time alloted to the timers, scan the thumbwheels.
20    The selectors stay on as placeholders, but scanning stops when
|     the timer stops. Otherwise, we would reset the count while outputting
|
|
```

```
|     the pulses, a very bad thing to do.
21
|
|
|
|   sel1    OneShot                                                    Seq1
22--| |-------|/|-----------------------------------------------------[LDHex]|
|  sel10    TenShot |
|---| |-------|/|---+
|
|
|
|     Hundreds is either on or off, so it only has two possible outcomes.
23    If selected, set the sequencer.
|
|
|
|  Sel100   Hundred   100Shot                                        Seq1:10
24--| |-------| |-------|/|-------------------------------------------[StepN]|
|
|
|
|     We have to turn on BR sometime. I chose to do it after the 1's
25    have been scanned.
|
|
|
|   sel1    OneShot                                                      BR
26--| |-------| |-----------------------------------------------------[Latch]|
|
|
|
|     The pulses would be free-running if we didn't specify exactly when
27    they're allowed. In the counting loop, when we're not scanning.
|     And when the pulse cycle is high, and there are pulse left to send.
|
|
| OneShot    Seq1:0   OffDelay                                          pulse
28--| |-------|/|-------|/|-------------------------------------------(OUT)|
| TenShot |  pulse  |
|---| |---+---| |---+
| 100Shot |
|---| |---+
|
|
|
|     As soon as the pulse goes high, start counting out the high
29    portion of the duty cycle. Reset the counter that controls the low
|     portion of the cycle.
|
|
```

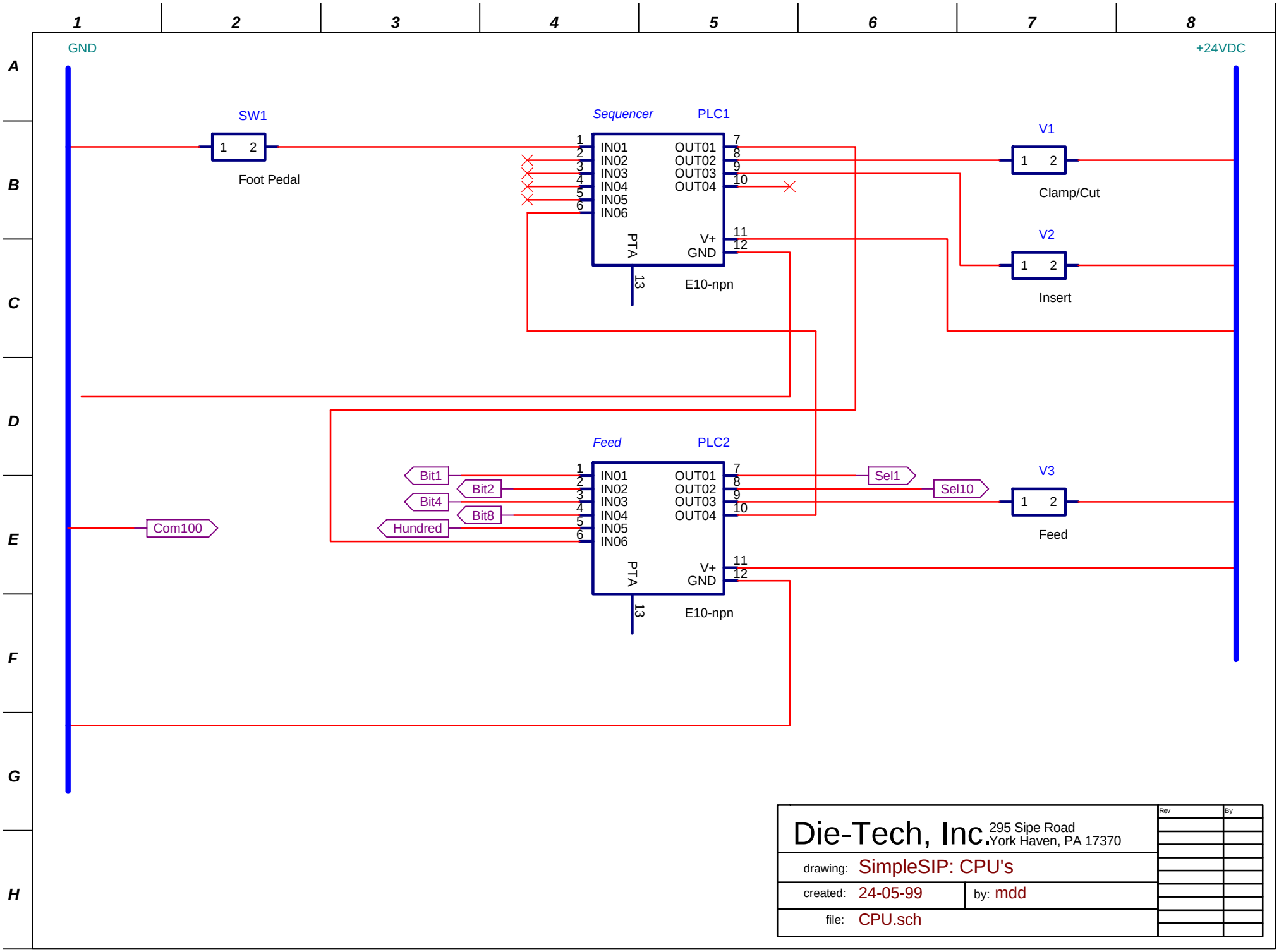
```
|     The OnDelay and OffDelay counters may be adjusted to suit your
30    application. I have set them to deliver the shortest reliable pulse
|     in my system, which is running air cylinders.
|
|
|  pulse    Clk:.01s                                                  OffDelay
31--| |-------| |-----------------------------------------------------------(CTR)|
|                                                                    | OnDelay
|                                                                   +-[RSctr]|
|
|
|
|     When the pulse goes low, start counting out the low portion of
32    the duty cycle.
|
|
|
| OffDelay  Clk:.01s                                                   OnDelay
33--| |-------| |-----------------------------------------------------------(CTR)|
|
|
|
|     When the low portion of the cycle is done, reset the counter
34    that controls the high portion.
|
|
|
| OnDelay                                                              OffDelay
35--| |-------------------------------------------------------------------[RSctr]|
|
|
|
|     This is the multiplier count. When counting the 10's pulses, each
36    pulse is counted here first. Then the output is sent to Seq1 every
|     10 pulses. The 100's count uses this counter 0 times.
|
|
|  sel10               pulse                                              10
37--| |-----------------| |-------------------------------------------------(CTR)|
|  Sel100   Hundred |
|---| |-------| |---+
|
|
|
|     This is where we control the pulse count. When 1's is in control,
38    the pulse directly decrements Seq1. When 10's is in charge, the
|     pulses go to the 10 counter, which then controls Seq1.
|
|
```

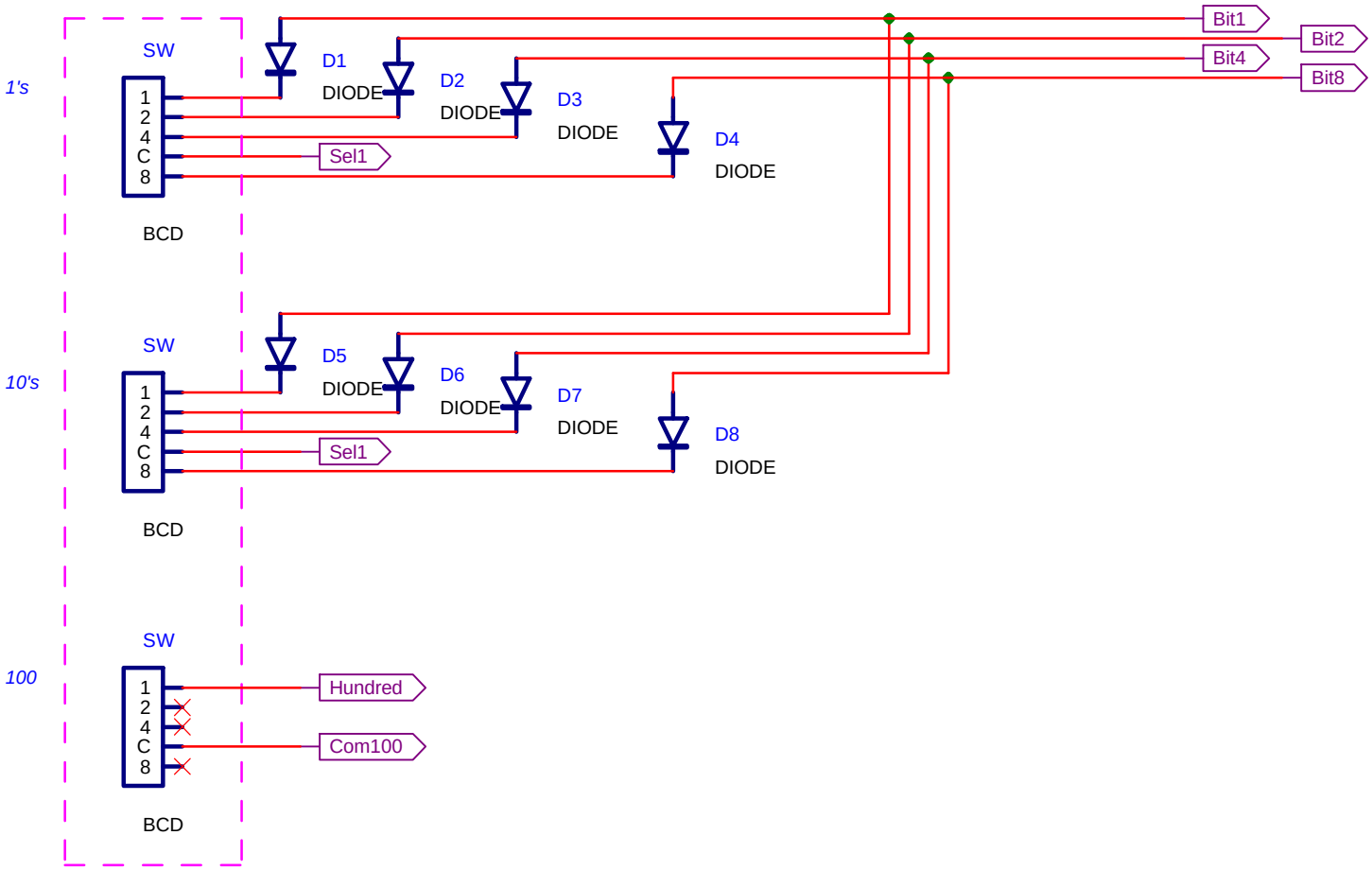
```
|   sel1     pulse                                                    Seq1
39--| |-------| |-------------------------------------------------------(CTR)|
|    10       sel1  |
|---| |-------|/|---+
|
|
|
|     If we're counting 10', and there is still count left on Seq1,
40    reset the 10 counter when it expires.
|
|
|
|  sel10       10                                                       10
41--| |-------| |-------------------------------------------------[RSctr]|
|  Sel100 |
|---| |---+
|
|
|
|     We've counted 1's, or they were set to zero. Set a coil so that
42    10's can start its scan. Exploiting resources here; the E10 has
|     2 "phantom" outputs, so why not put them to work. Saves relays.
|
|
|  Seq1:0   OneShot    pulse                                           exp1
43--| |-------| |-------|/|-----------------------------------------------(OUT)|
|
|
|
|     Done with 10's or they were set to zero. Set a coil so that
44    100's can start. Using another "phantom" here.
|
|
|
|  Seq1:0   TenShot    pulse                                           exp2
45--| |-------| |-------|/|-----------------------------------------------(OUT)|
|
|
|
|     All done with pulses. Signal the other PLC by clearing BR.
46
|
|
|
|  Seq1:0   100Shot    pulse                                             BR
47--| |-------| |-------|/|-------------------------------------------[Clear]|
|
|
|
|     END
48
|
|
|
```

```
|
49
|
|
|
```

Combined I/O Definition Table

I/O#	Input	Output	Relay	Timer	Value	Counter	Value
1	bit1	sel1	Sel100	OneShot	1	Seq1	0
2	bit2	sel10		TenShot	1	10	9
3	bit4	pulse		100Shot	1	OnDelay	9
4	bit8	BR				OffDelay	9
5	Hundred	exp1					
6	MA	exp2					
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							


1
2
3
4
5
6
7
8
A
B
C
D
E
F
G
H
GND
+24VDC
SW1
1 2
Foot Pedal
Sequencer
PLC1
IN01
IN02
IN03
IN04
IN05
IN06
PTA
OUT01
OUT02
OUT03
OUT04
V+
GND
E10-npn
V1
1 2
Clamp/Cut
V2
1 2
Insert
Feed
PLC2
Bit1
Bit2
Bit4
Bit8
Hundred
Com100
Sel1
Sel10
V3
1 2
Feed
Die-Tech, Inc.
295 Sipe Road
York Haven, PA 17370
drawing: SimpleSIP: CPU's
created: 24-05-99
by: mdd
file: CPU.sch
Rev
By



Die-Tech, Inc. 295 Sipe Road York Haven, PA 17370		Rev	By
drawing: Simple SIP: Thumbwheel Assy			
created: 24-05-99	by: m. dennis		
file: Thumbwheels.sch			