

start

ST = 1000000

clock = 0

SOStar = 0

SOCircle = 0

SOTriangle = 0

} for computing the average length of queues

NoWS = 0 (number of waiting stars)

NoWC = 0 ( ~ ~ ~ circles)

NoWT = 0 ( ~ ~ ~ triangles)

PCSQ = 0 (Previous clock stars queue)

PCTQ = 0 ( ~ ~ triangle ~ )

PCCQ = 0 ( ~ ~ circle ~ )

SOS1 = 0 (sum of service 1)

SOS2 = 0 (sum of service 2)

S1 = False

S2 = False

PCSI = 0 (Previous clock of server 1)

PCS2 = 0 ( ~ ~ ~ 2)

NoD (Number of discarded stars)

NoES (number of entered stars)

SOAT (sum of turnaround time for circle triangle in S2)

SOCTC (sum of circle triangle which get complete service from

QrMax = input() (gets input from user) S2)

Qr = NULL (Queue <Integer>)

ETQB = 0 (entered time to Qr buffer (used for <sup>turnaround time</sup> preempted circle triangles and for turnaround time of it))

2nd

PreEmp = False

RTPE = 0 (remaining time for preempted circleTriangles)

SOBS1 = 0 (sum of blocking time for S1)

PCBS1 = 0 (previous clock blocking time S1)

tbl[0] = StarEnter()

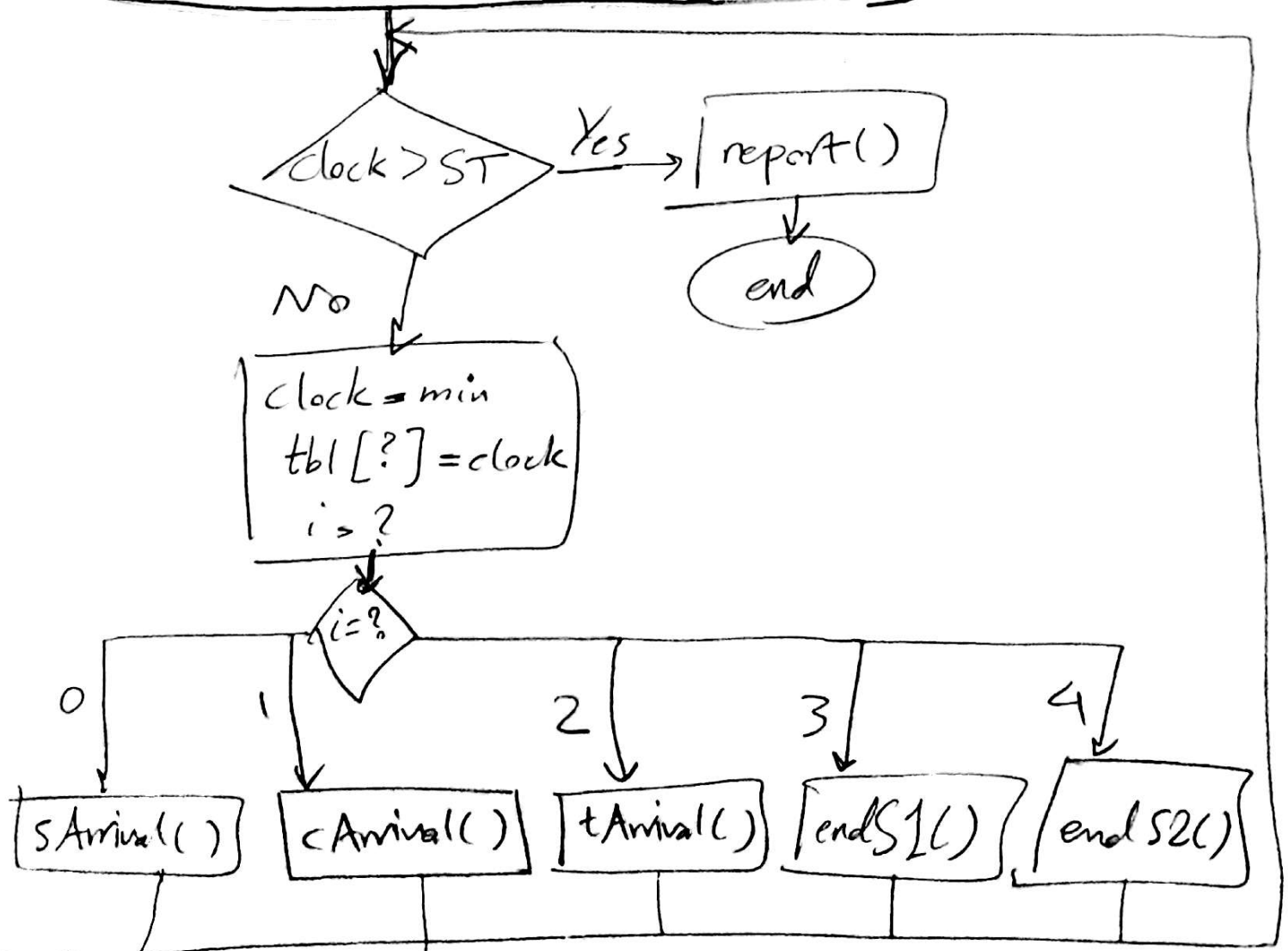
tbl[1] = CircleEnter()

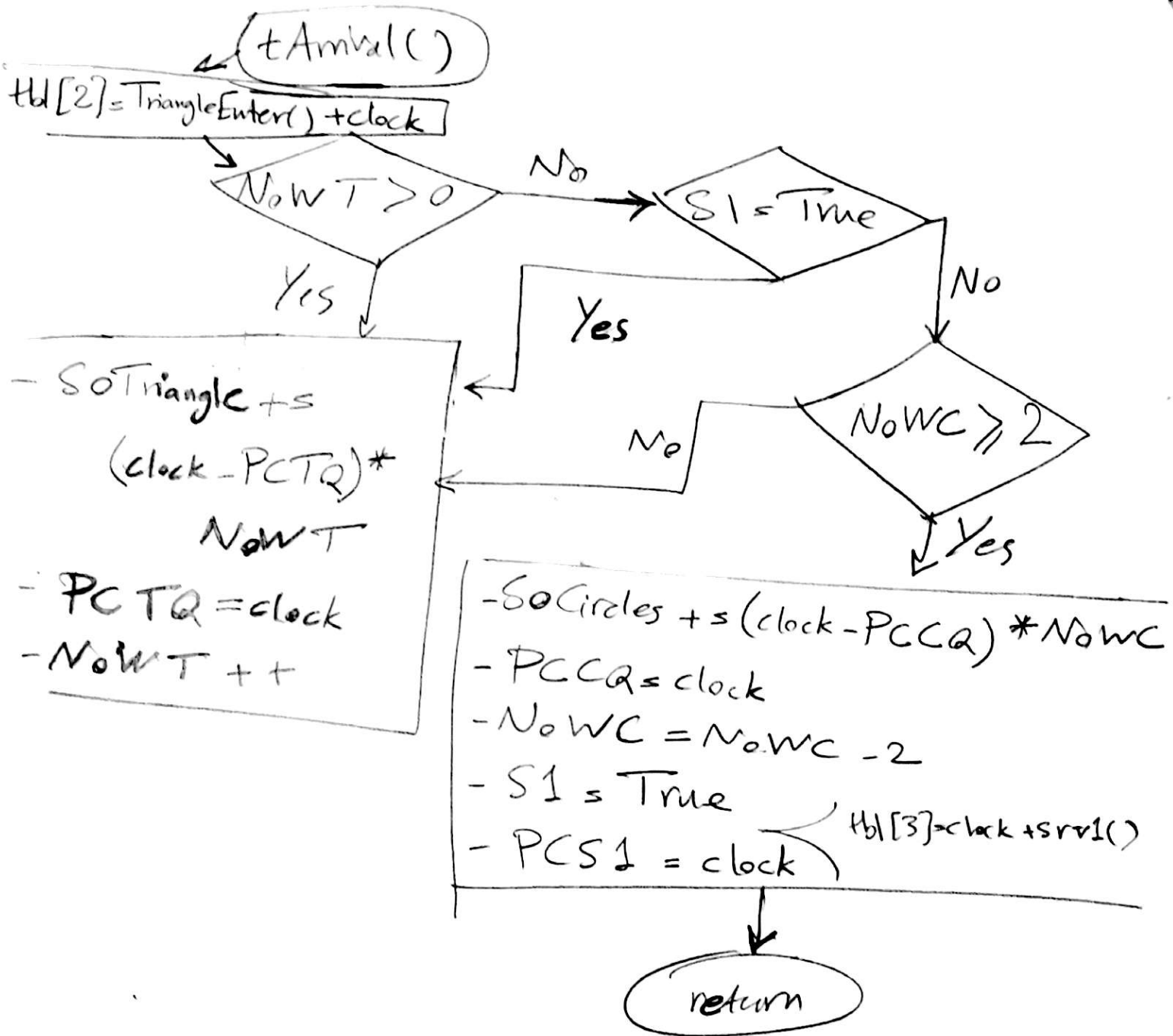
tbl[2] = TriangleEnter()

tbl[3] = 1000001 → end of service 1

tbl[4] = 1000001 → end of service 2

block = False / STW = 0 (service to who?) (0 → CircleTriangle)  
1 → Star  
for server2





Arrival()

$tbl[1] = \text{CircleEnter}() + \text{clock}$

$\text{NowC} \geq 2$

No

$S1 = \text{True}$

Yes

No

-  $\text{SOCircles} += (\text{clock} - \text{PCCQ})$   
\*  $\text{NowC}$

-  $\text{PCCQ} = \text{clock}$

-  $\text{NowC}++$

No

$\text{NowT} > 0$

Yes

-  $\text{SoTriangles} += (\text{clock} - \text{PCTQ})$   
\*  $\text{NowT}$

-  $\text{PCTQ} = \text{clock}$

-  $\text{NowT}--$

-  $\text{NowC}--$

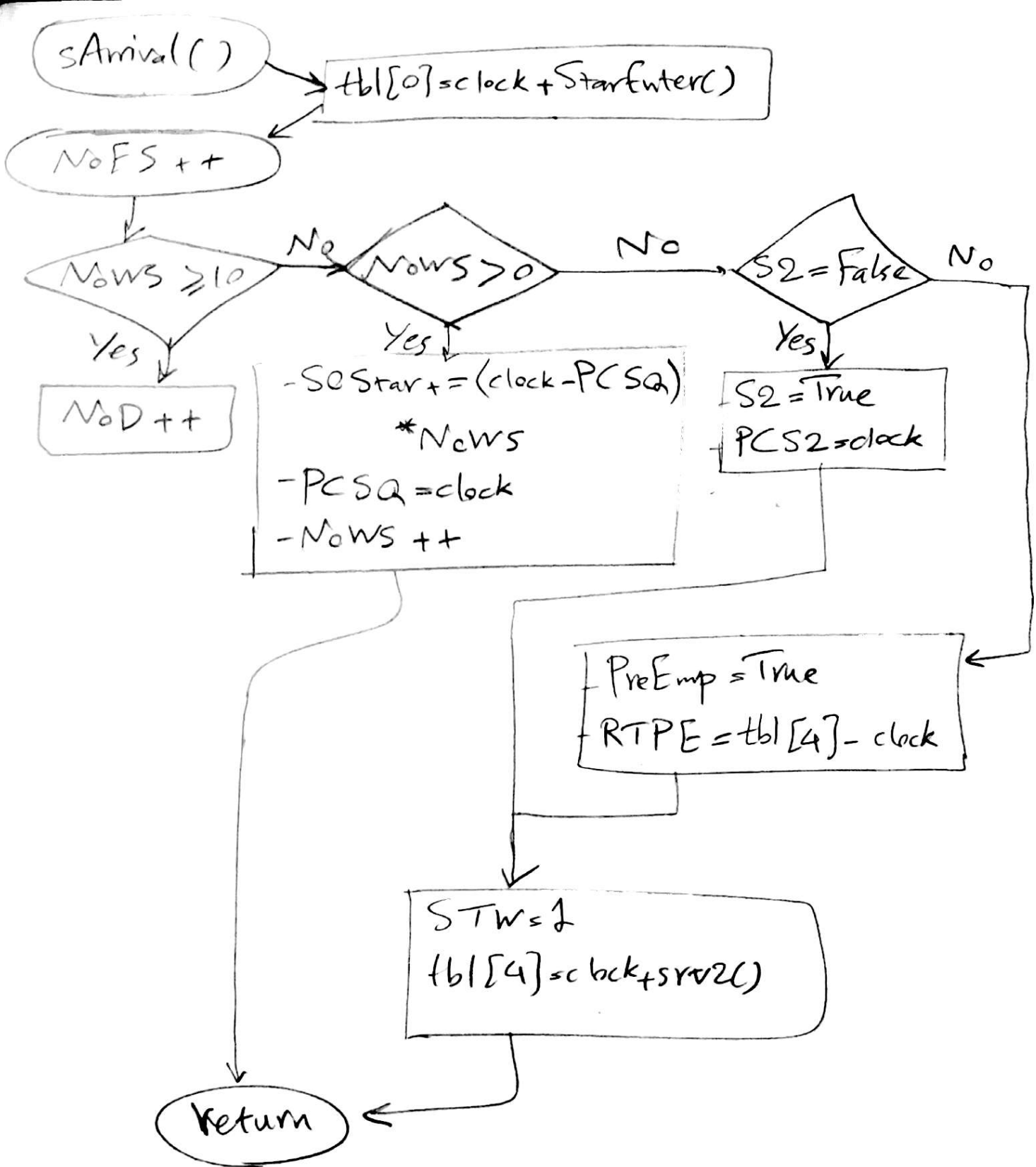
-  $S1 = \text{True}$

-  $\text{PCS1} = \text{clock}$

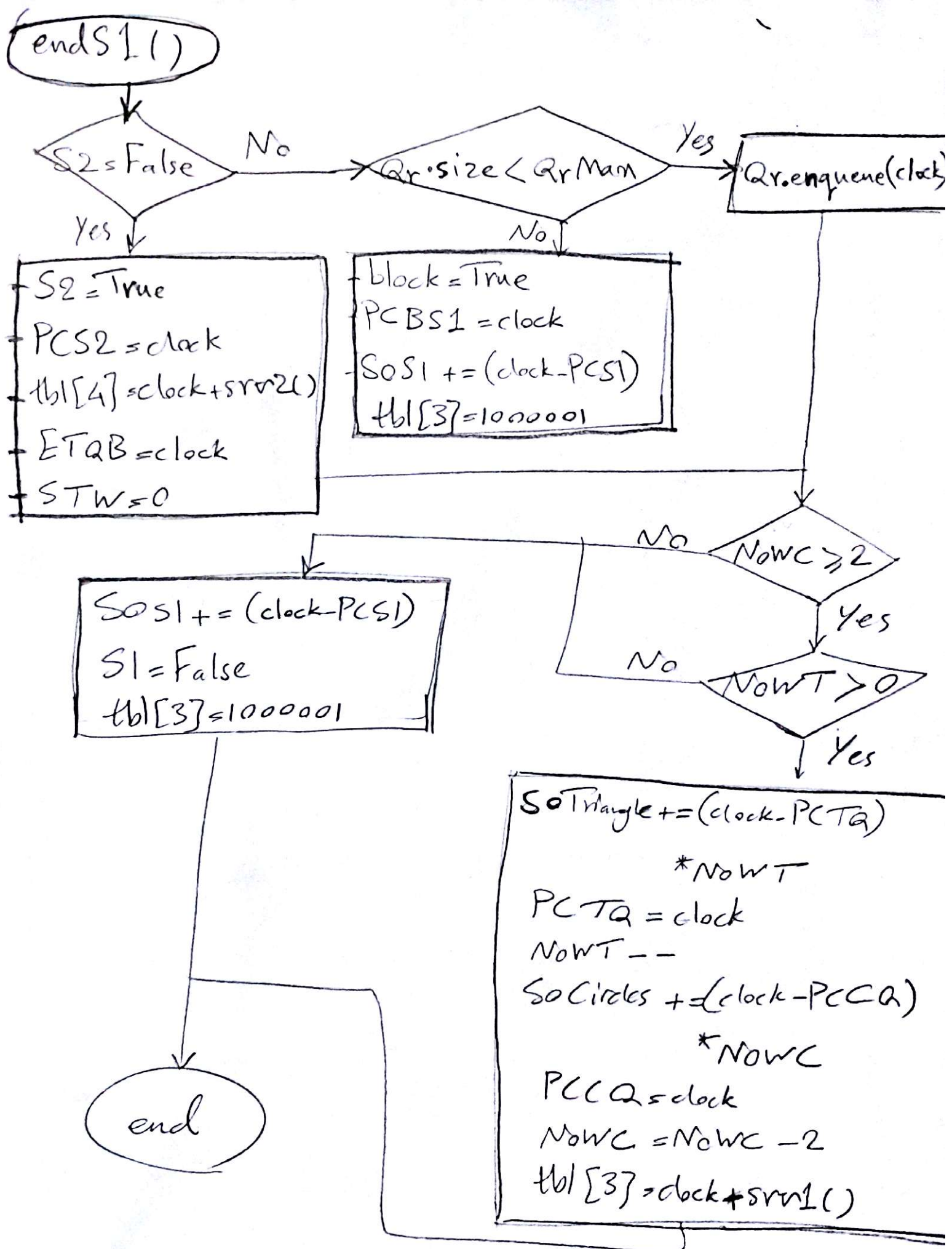
$tbl[3] = \text{clock} + \text{srv1}()$

return

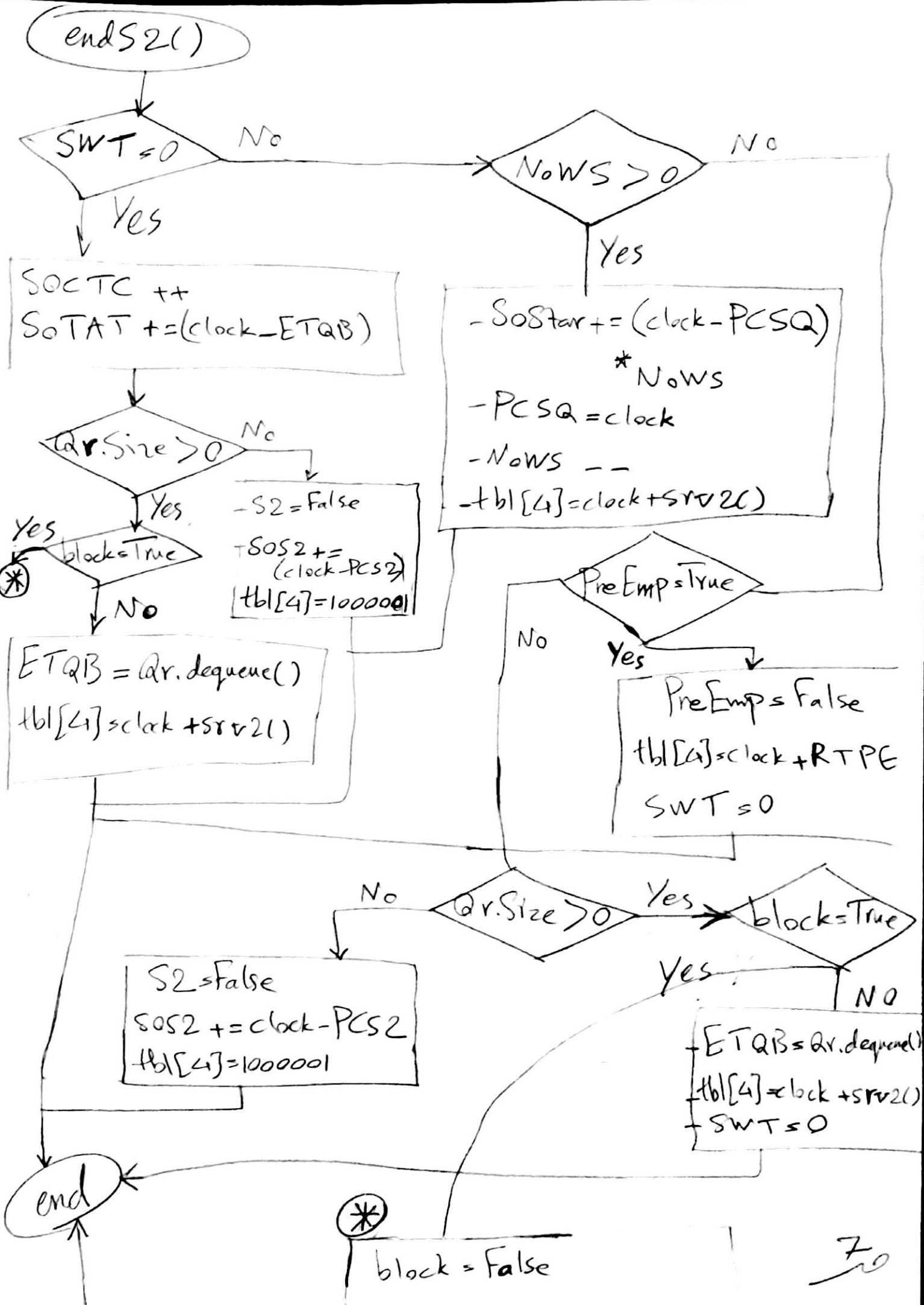
40



5.0



6



70



قبل

ETQB = Qr. dequeue()

SWT = 0

SOBS1 += clock - PCBS1

~~PCS1 = clock~~

Qr. enqueue(clock)

end

No  
 $NowC \geq 2$

Yes

No

$NowT > 0$

Yes

S1 = False

PCS1 = clock

SOTriangle += (clock - PCTa)

\* NowT

PCTa = clock

NowT --

SOCircles += (clock - PCCQ)

\* NowC

PCCQ = clock

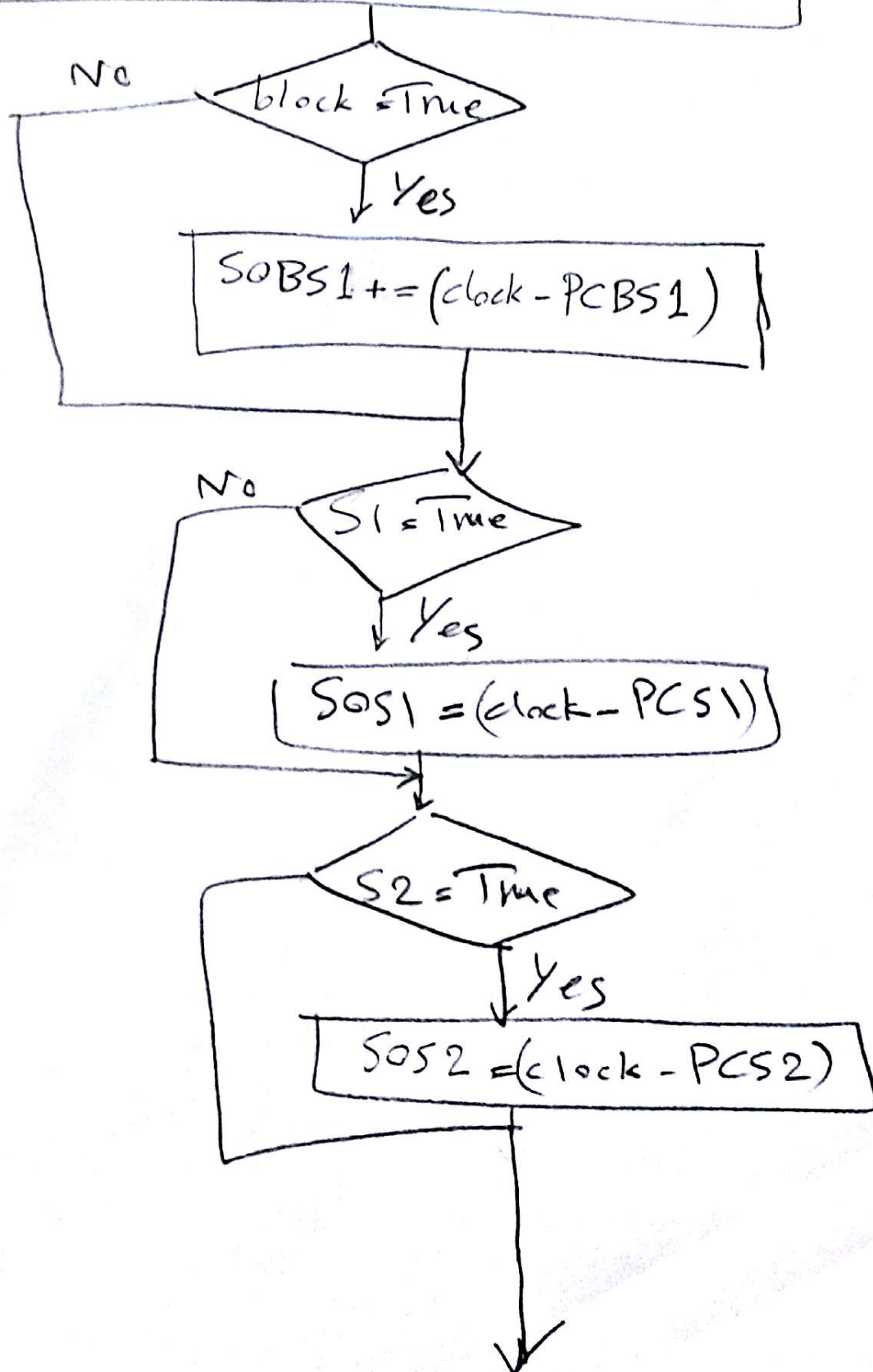
NowC = NowC - 2

tbl[3] = clock + srv1()

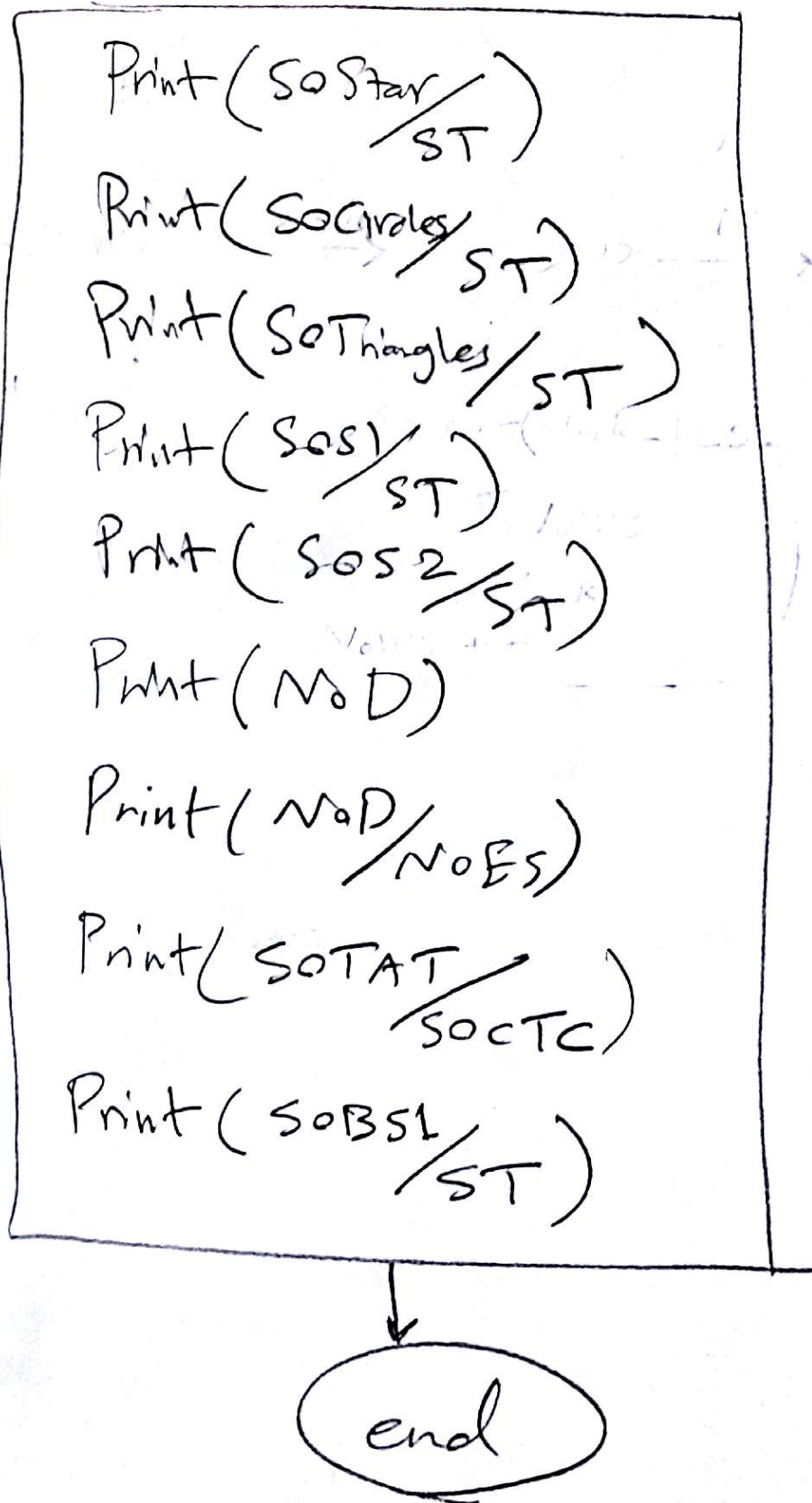


report(c)

$SOStar += (clock - PCSQ) * NOWS$   
 $SOCircles += (clock - PCCQ) * NOWC$   
 $SOTriangles += (clock - PCTQ) * NOWT$



20



low