Embedded Systems

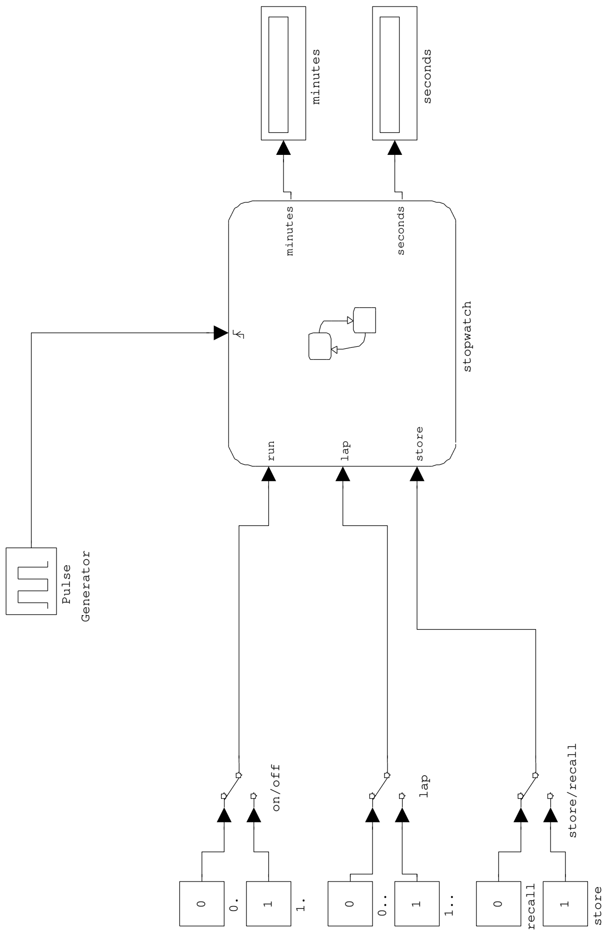
Assignment 4

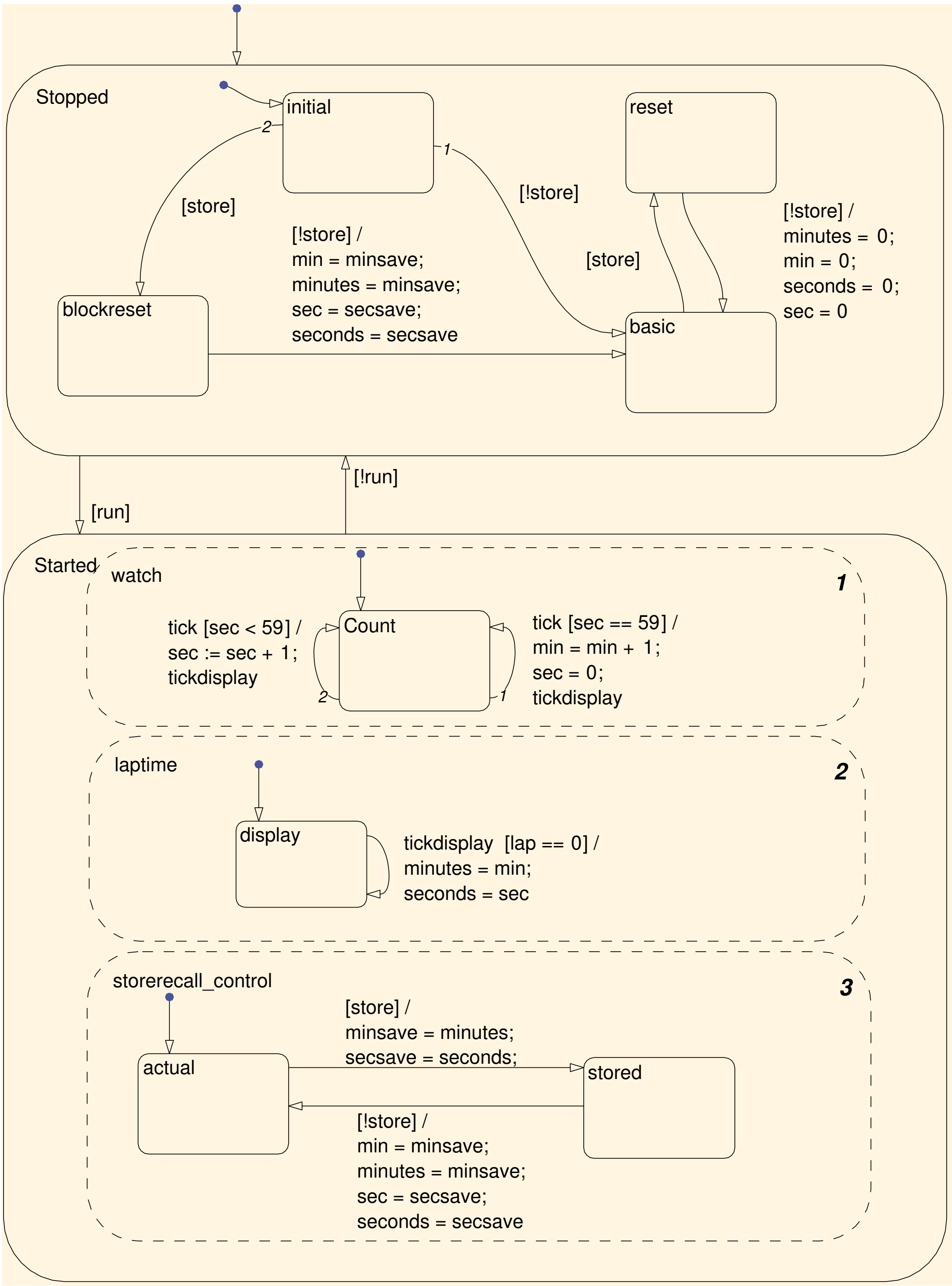
Jan Hendrik Dithmar
2031259

Pascal Gwosdek
2505221

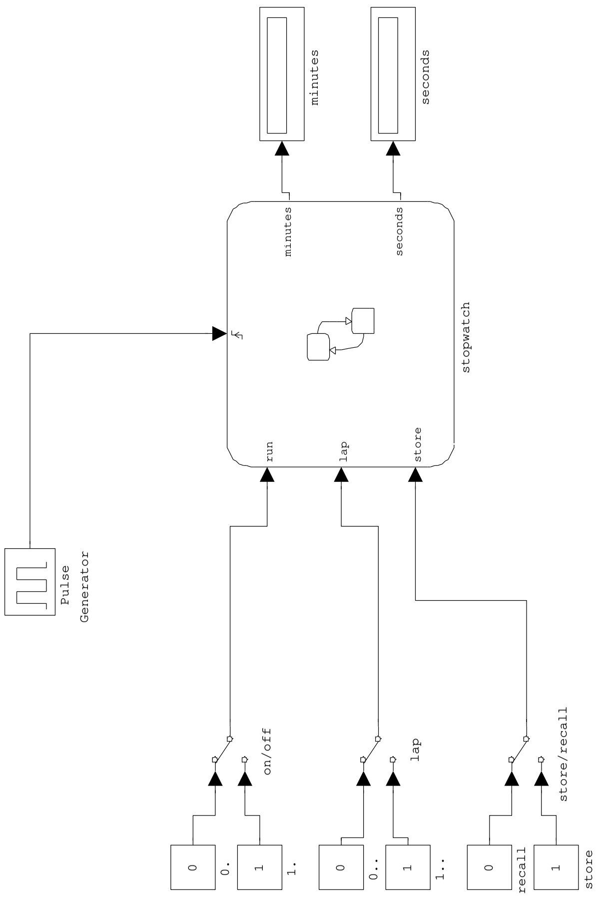
We did Exercise 1, of course. For exercise 2, we combined the first three parts in one result and the last part in another result, because there was a change in the way we handeled time.

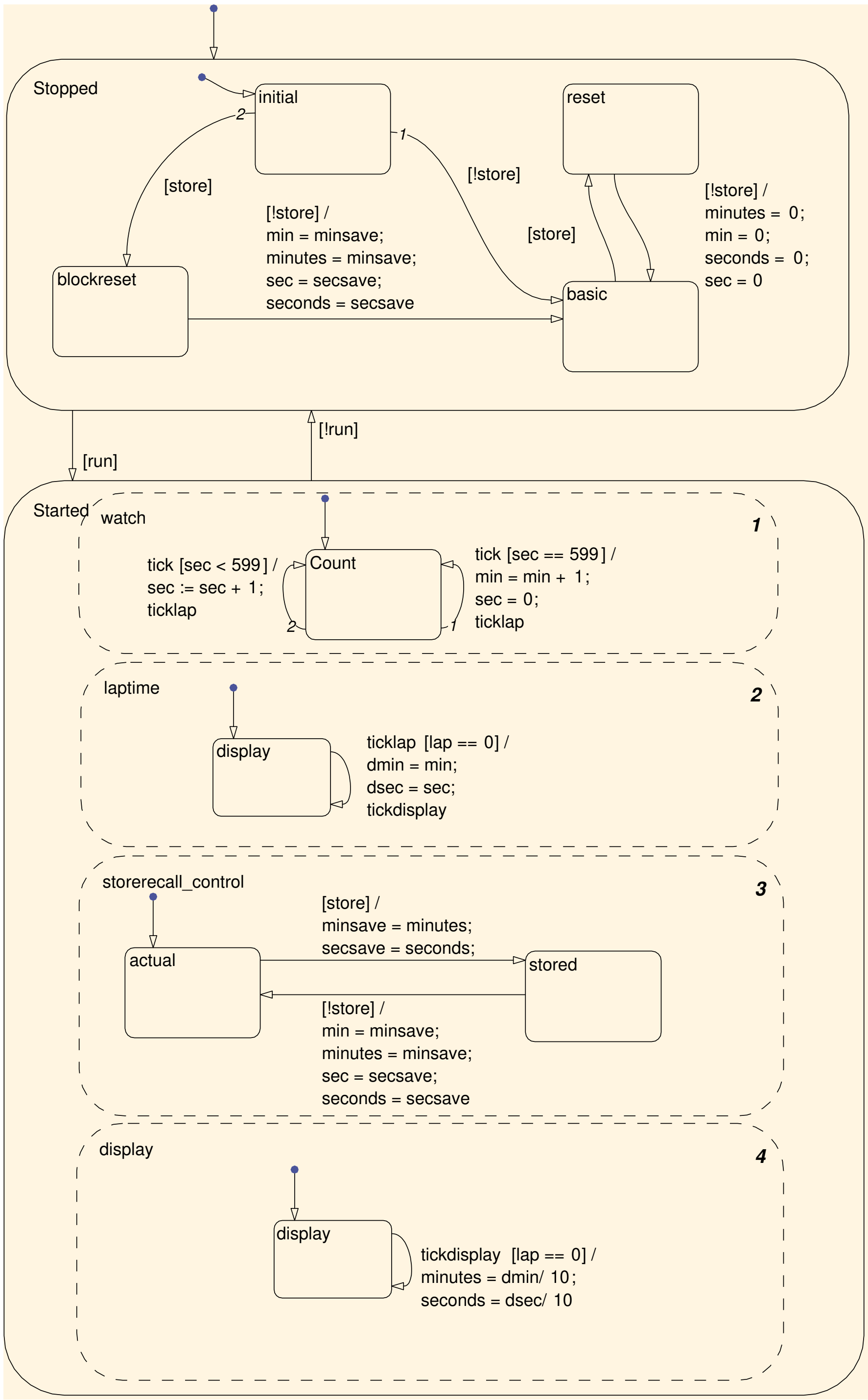
Exercise 2, parts 1, 2 and 3:




Stopped
initial
reset
2
1
[store]
[!store]
[!store] /
min = minsave;
minutes = minsave;
sec = secsave;
seconds = secsave
[store]
[!store] /
minutes = 0;
min = 0;
seconds = 0;
sec = 0
blockreset
basic
[!run]
[run]
Started
watch
1
tick [sec < 59] /
sec := sec + 1;
tickdisplay
Count
tick [sec == 59] /
min = min + 1;
sec = 0;
tickdisplay
laptime
2
display
tickdisplay [lap == 0] /
minutes = min;
seconds = sec
storerecall_control
3
[store] /
minsave = minutes;
secsave = seconds;
actual
stored
[!store] /
min = minsave;
minutes = minsave;
sec = secsave;
seconds = secsave

Exercise 2, part 4:




Stopped
initial
reset
2
1
[store]
[!store]
[store]
[!store] /
min = minsave;
minutes = minsave;
sec = secsave;
seconds = secsave
[!store] /
minutes = 0;
min = 0;
seconds = 0;
sec = 0
blockreset
basic
[run]
[!run]
Started
watch
1
tick [sec < 599] /
sec := sec + 1;
ticklap
Count
tick [sec == 599] /
min = min + 1;
sec = 0;
ticklap
laptime
2
display
ticklap [lap == 0] /
dmin = min;
dsec = sec;
tickdisplay
storerecall_control
3
[store] /
minsave = minutes;
secsave = seconds;
actual
stored
[!store] /
min = minsave;
minutes = minsave;
sec = secsave;
seconds = secsave
display
4
display
tickdisplay [lap == 0] /
minutes = dmin/ 10;
seconds = dsec/ 10