Finite statemachines (FSM)

Tilstandsmaskiner

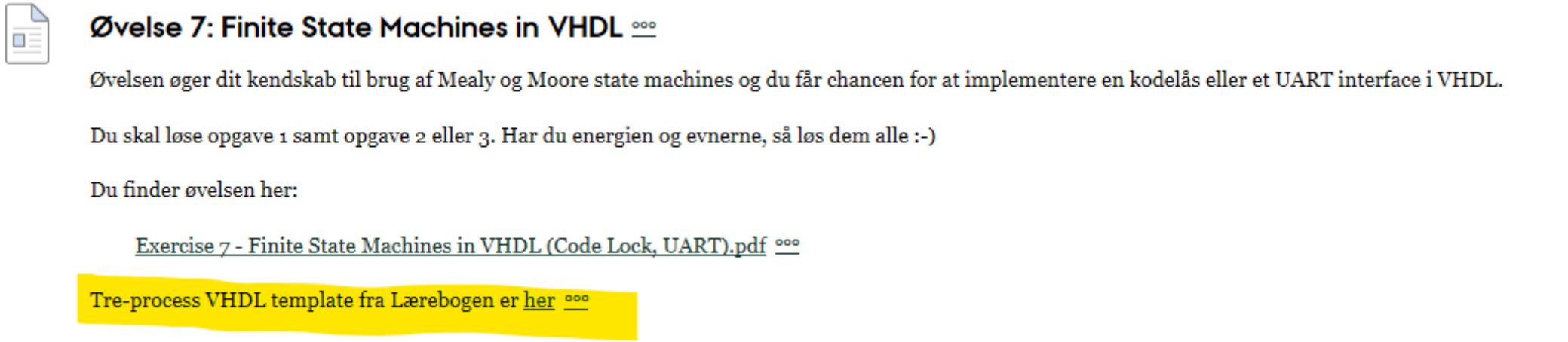
Øvelse 7

Agenda

- VHDL template
- Anbefaling vedr. code-lock opgaven
- Hvordan skal tilstandsdiagrammet for Code-lock opgaven tolkes
- Anbefaling Code-lock vs. UART

VHDL template

- Der ligger en VHDL template på Blackboard



Øvelse 7: Finite State Machines in VHDL

Øvelsen øger dit kendskab til brug af Mealy og Moore state machines og du får chancen for at implementere en kodelås eller et UART interface i VHDL.

Du skal løse opgave 1 samt opgave 2 eller 3. Har du energien og evnerne, så løs dem alle :-)

Du finder øvelsen her:

Exercise 7 - Finite State Machines in VHDL (Code Lock, UART).pdf

Tre-process VHDL template fra Lærebogen er her

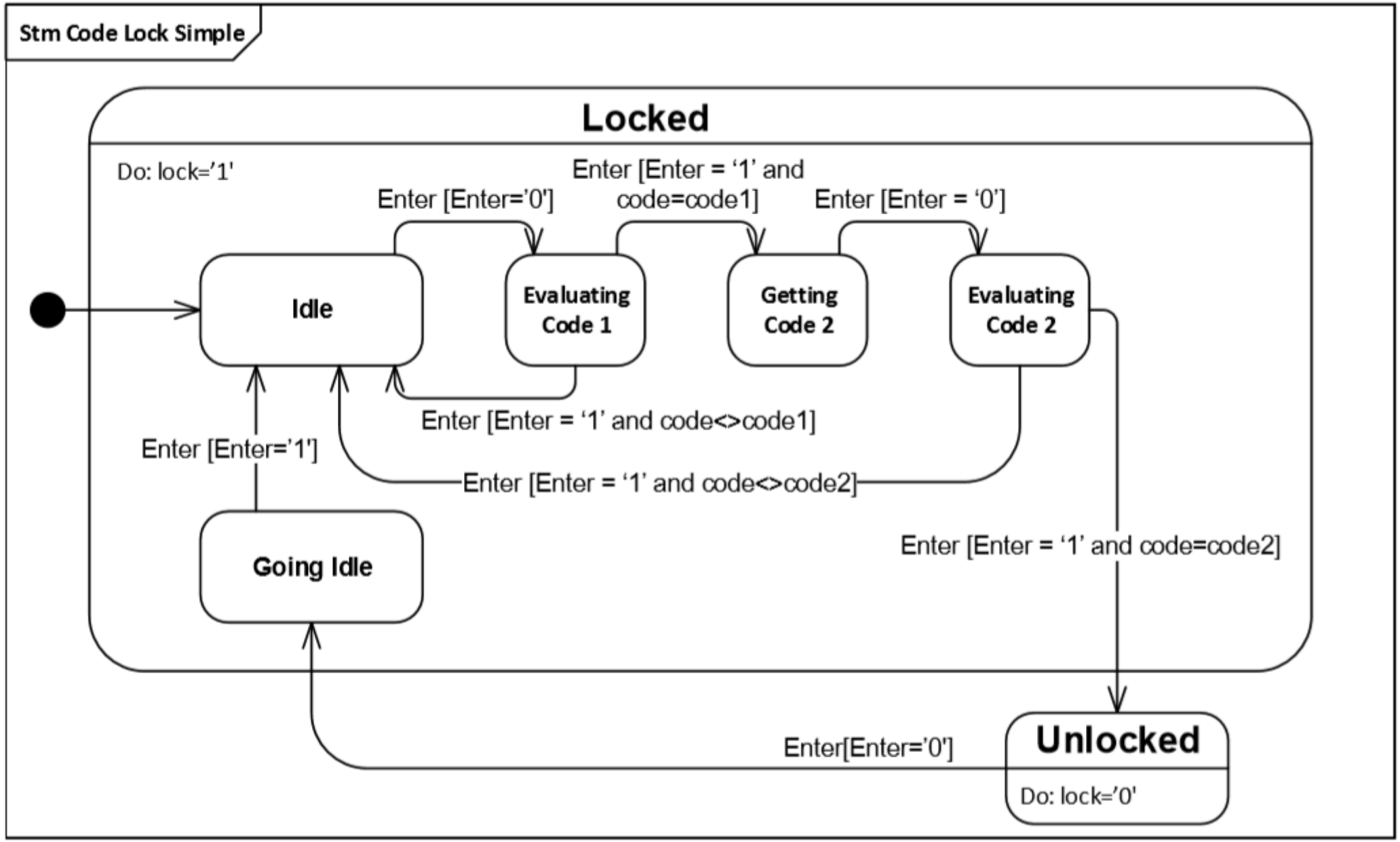


Fig. 4: *Code lock simple STM.*

Anbefaling vedr. Code-lock opgaven

Hierakisk FSM

```
architecture rtl of code_lock_fsm is

    -- Indre statemachine, eksekveret i "Locked" state
    type state_type_inner is (idle,ev_c1,get_c2,ev_c2, going_idle);
    -- Register to hold the current state
    signal present_state_inner,next_state_inner : state_type_inner;

    -- Ydre statemachine med "Locked" og "Unlocked" states
    type state_type_outer is (locked,unlocked);
    signal present_state_outer, next_state_outer : state_type_outer;
```

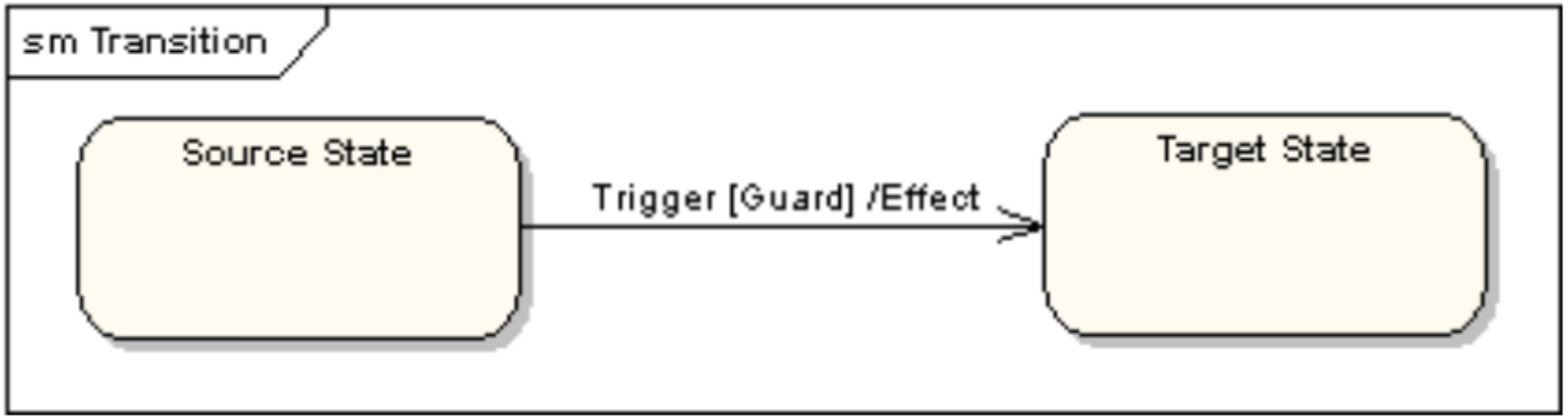
Enkelt FSM

```
architecture rtl of code_lock_fsm is

    -- Indre statemachine, eksekveret i "Locked" state
    type state_type is (idle,ev_c1,get_c2,ev_c2, going_idle, unlocked);
    -- Register to hold the current state
    signal present_state,next_state : state_type;
```

Hvordan skal tilstandsdiagrammet for code-lock opgaven tolkes

SysML og statediagram



"Trigger" is the cause of the transition, which could be a signal, an event, a change in some condition, or the passage of time. **"Guard"** is a condition which must be true in order for the trigger to cause the transition. **"Effect"** is an action which will be invoked directly on the object that owns the state machine as a result of the transition.

Hvordan forstår man så det her statediagram?

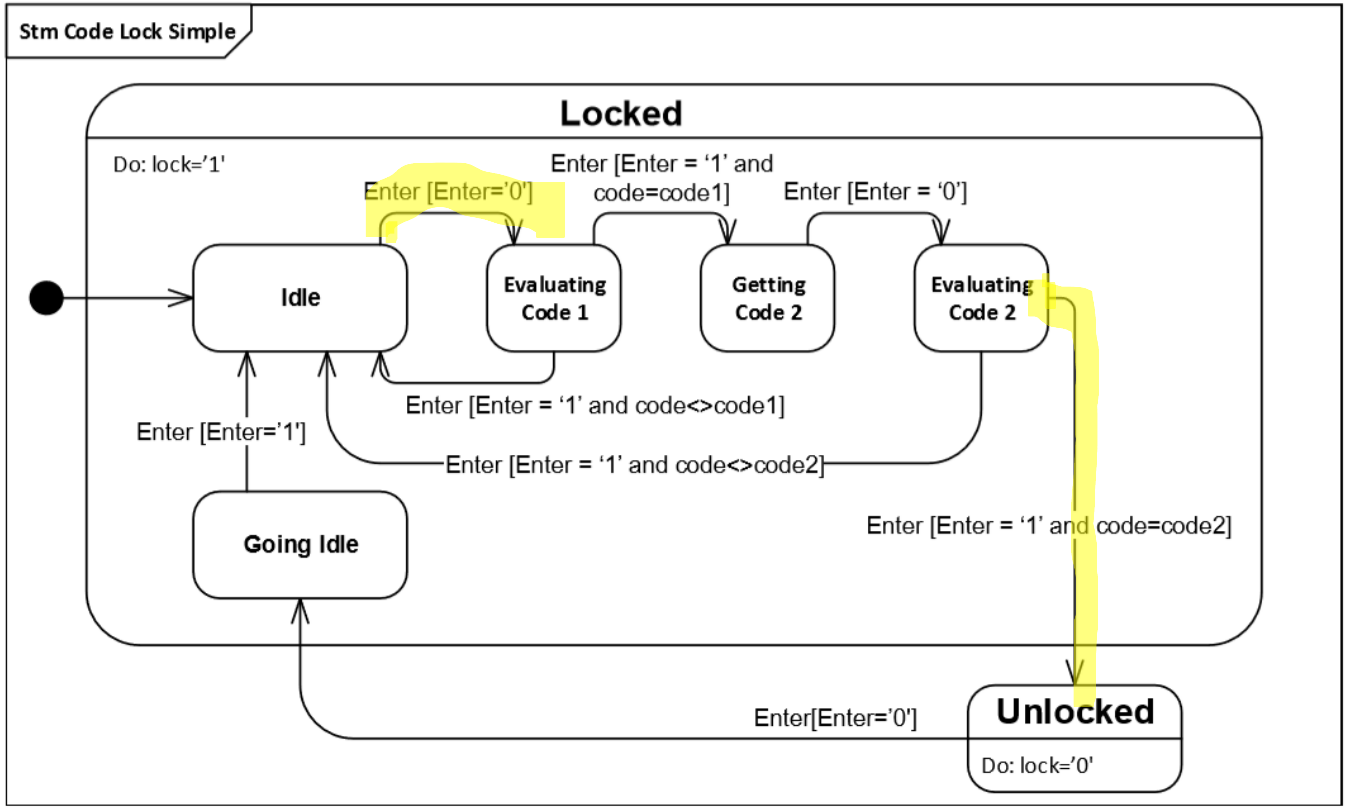


Fig. 4: *Code lock simple STM.*

Sådan her….

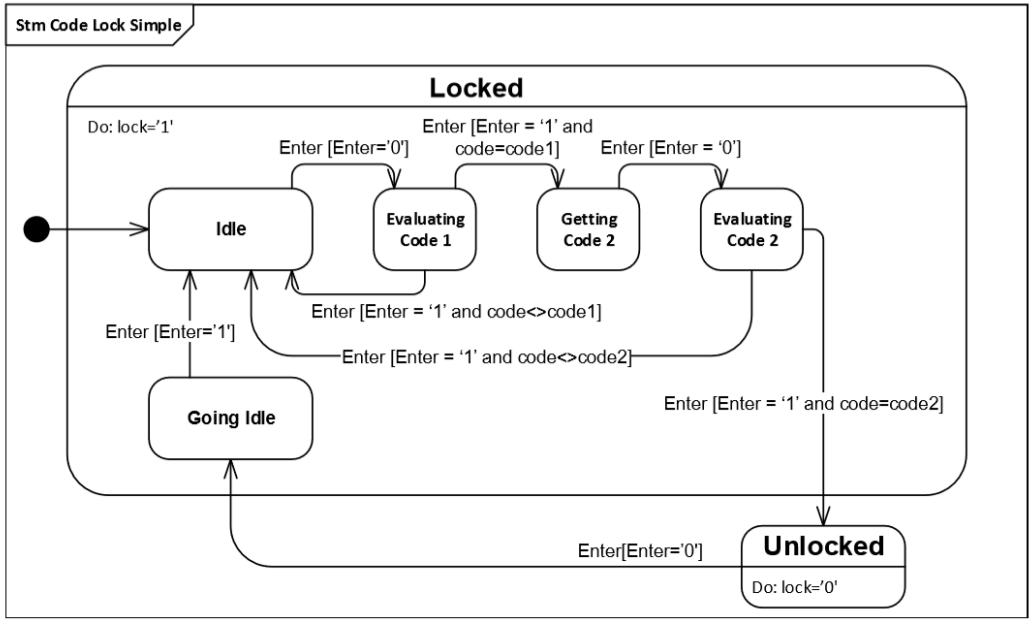
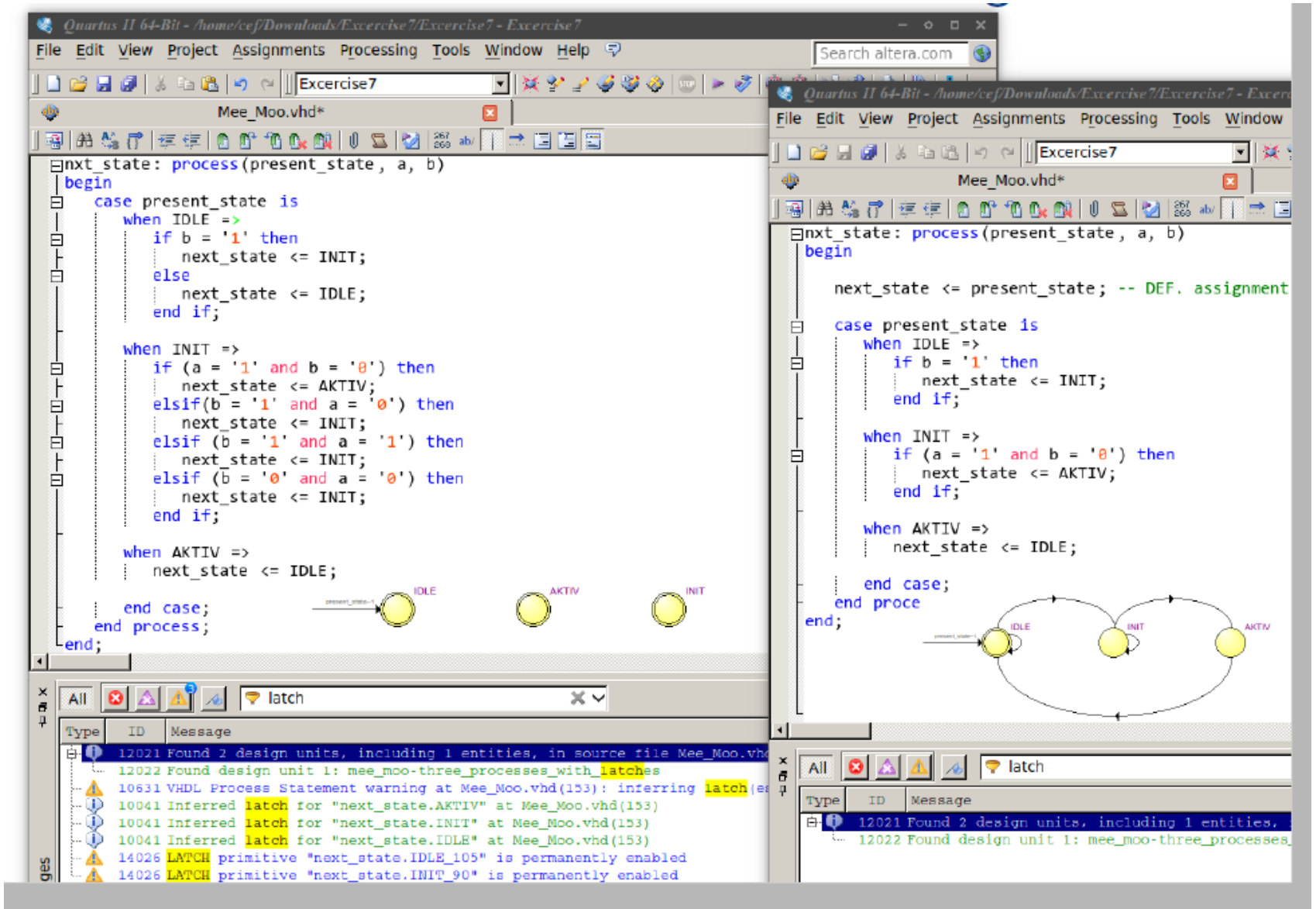


Fig. 4: *Code lock simple STM.*

- KEY tryk-hold, KEY slip, KEY tryk-hold, KEY slip, KEY tryk-hold, KEY slip…
  - Og teste Enter=0, Enter=1,…

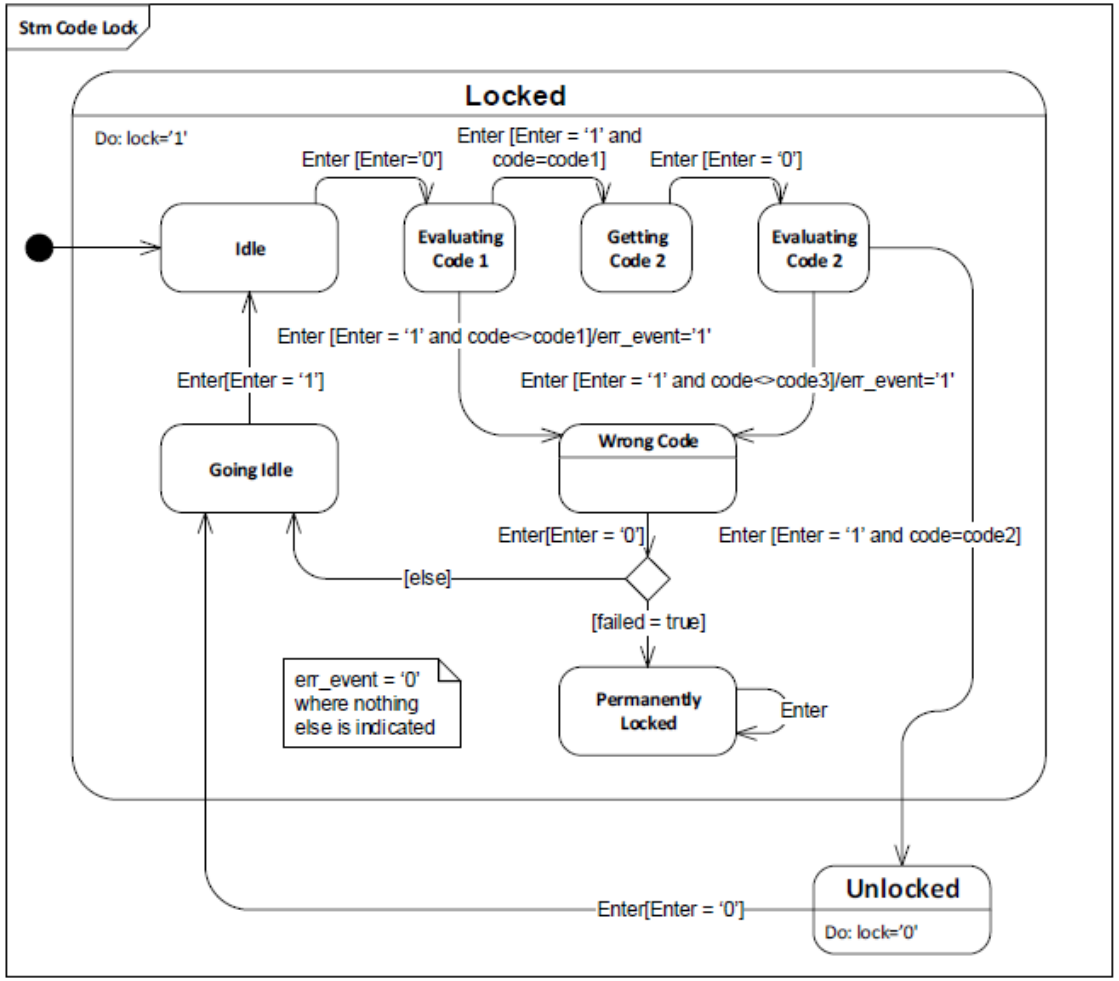
Utilsigtede latches

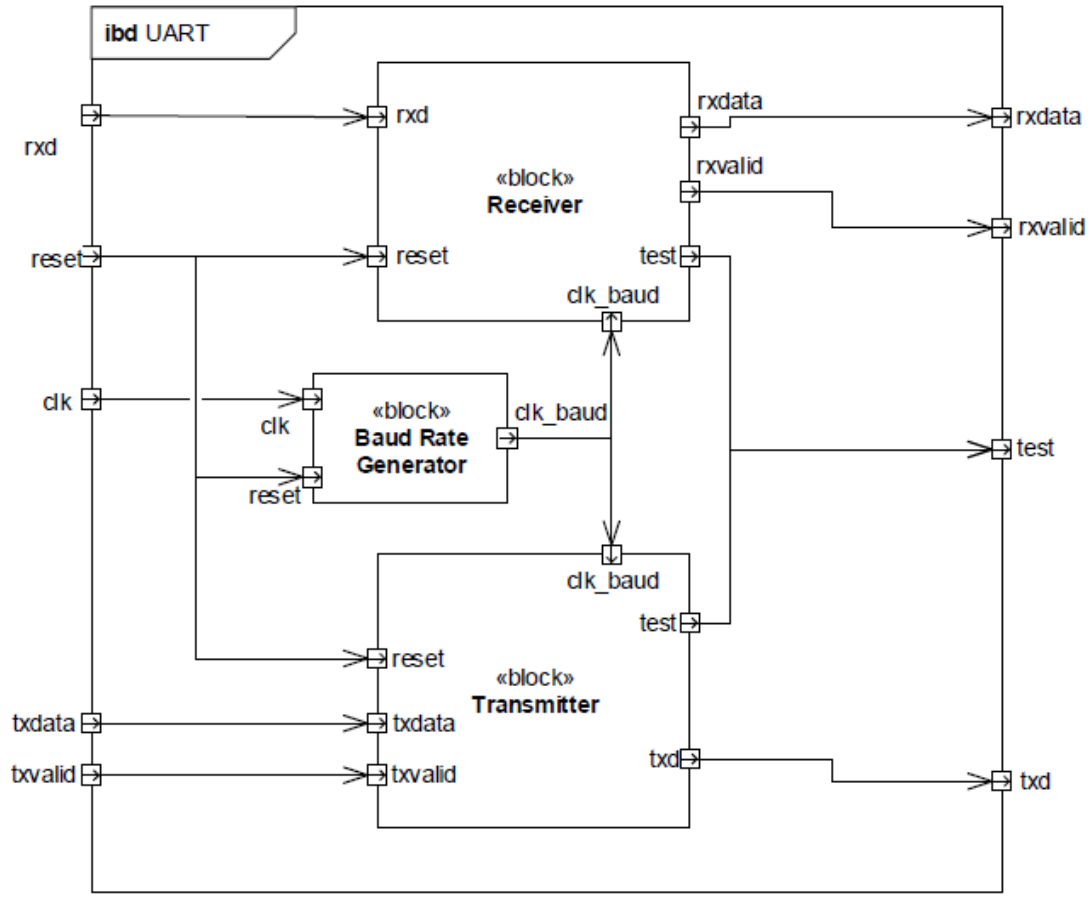


Fejlsøgningsmetodik

Formel procedure: hvad hjælpelærer vil spørge om

1. Hvad virker ikke?
2. Hvor kan fejlen isoleres til?
3. Hvordan vil du kunne teste at det virker?
4. (Hvad skal ellers undersøges for at rette fejlen?)


Stm Code Lock
Locked
Do: lock='1'
Enter [Enter='0']
Enter [Enter = '1' and code=code1]
Enter [Enter = '0']
Idle
Evaluating Code 1
Getting Code 2
Evaluating Code 2
Enter [Enter = '1' and code<>code1]/err_event='1'
Enter[Enter = '1']
Enter [Enter = '1' and code<>code3]/err_event='1'
Going Idle
Wrong Code
Enter[Enter = '0']
Enter [Enter = '1' and code=code2]
[else]
[failed = true]
err_event = '0' where nothing else is indicated
Permanently Locked
Enter
Unlocked
Do: lock='0'
Enter[Enter = '0']


ibd UART
rxd
rxdata
rxvalid
«block»
Receiver
reset
test
clk_baud
clk
«block»
Baud Rate Generator
txdata
txvalid
«block»
Transmitter
txd