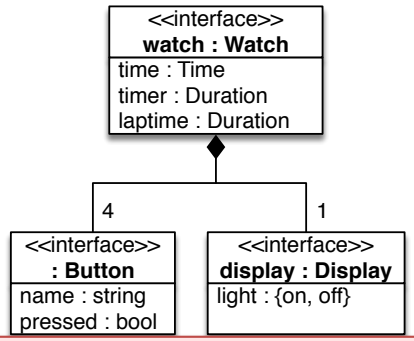
# CS445 / ECE451 / CS645 / SE463

## Software Requirements Specification & Analysis

# Hierarchical Machines

# Stopwatch Example




**Event Macros**

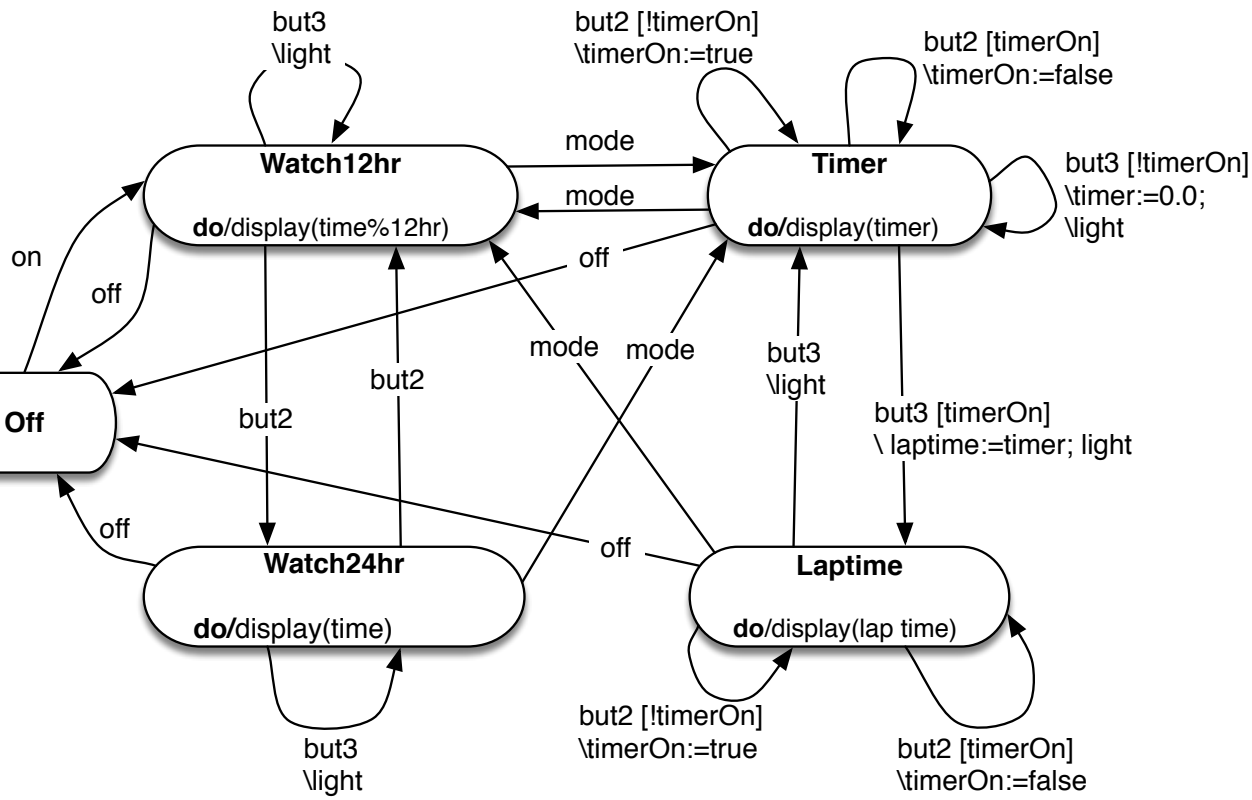
on = when ( On/Off:Button.pressed )
off = when ( On/Off:Button.pressed )
mode = when ( mode:Button.pressed )
but2 = when ( but2:Button.pressed )
but3 = when ( but3:Button.pressed )

**Domain Model Variable Macros**

time = watch.time
timer = watch.timer
laptime = watch.laptime
light = display.light

**Vars**

timerOn: bool




ay Godfrey)

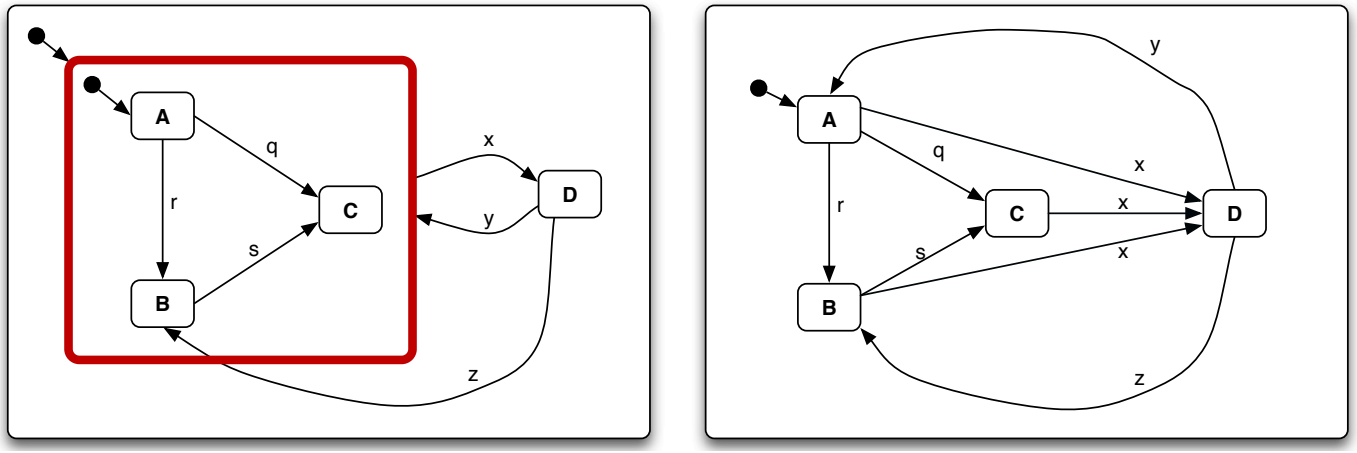
# Enrichments

- Generally, Mealy machines in practice result in large models with lots of criss-crossing transitions
- Three language features to ease complexity
  - Hierarchy
  - Concurrency
  - Inter-machine Communication

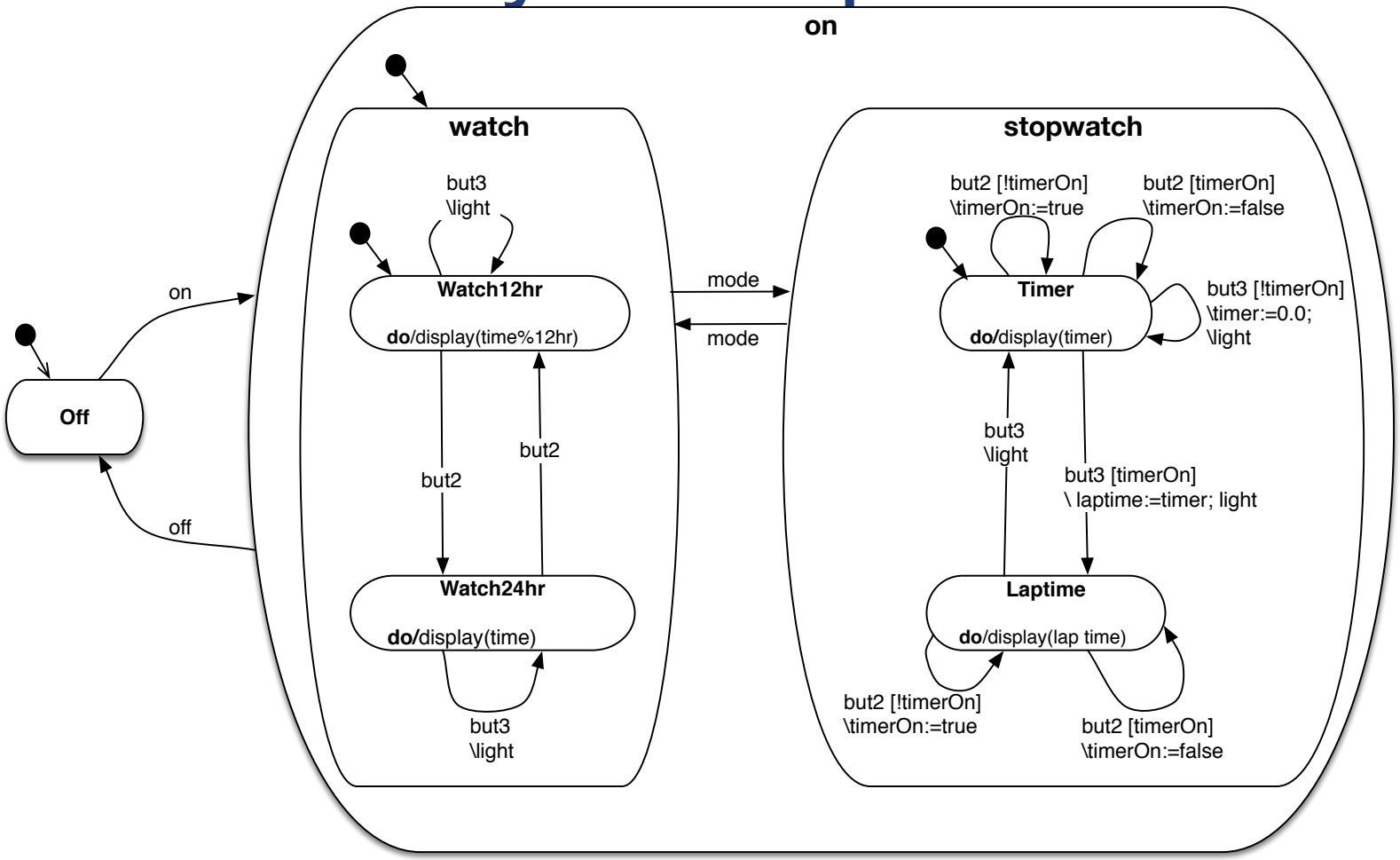
# Hierarchical states

Hierarchy is used to cluster states that have some similar behaviours / exiting transitions.

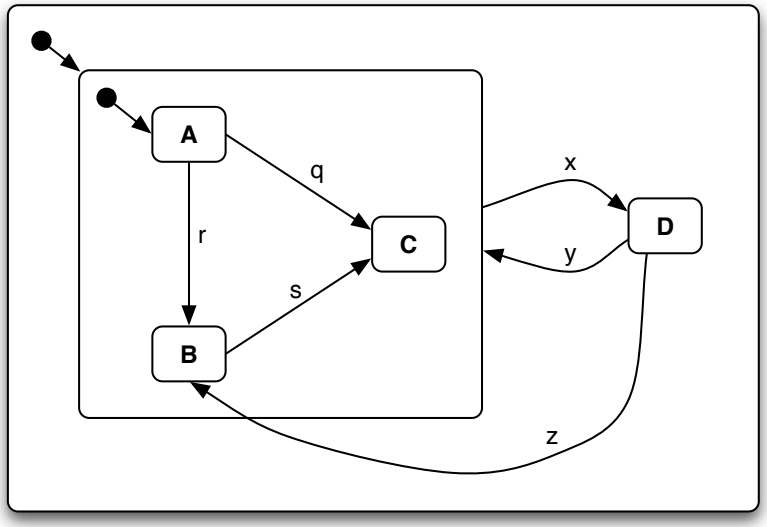
- One transition leaving a superstate represents a transition from each of the superstate's descendent states.



# Hierarchy in Stopwatch



# Hierarchy (semantics)



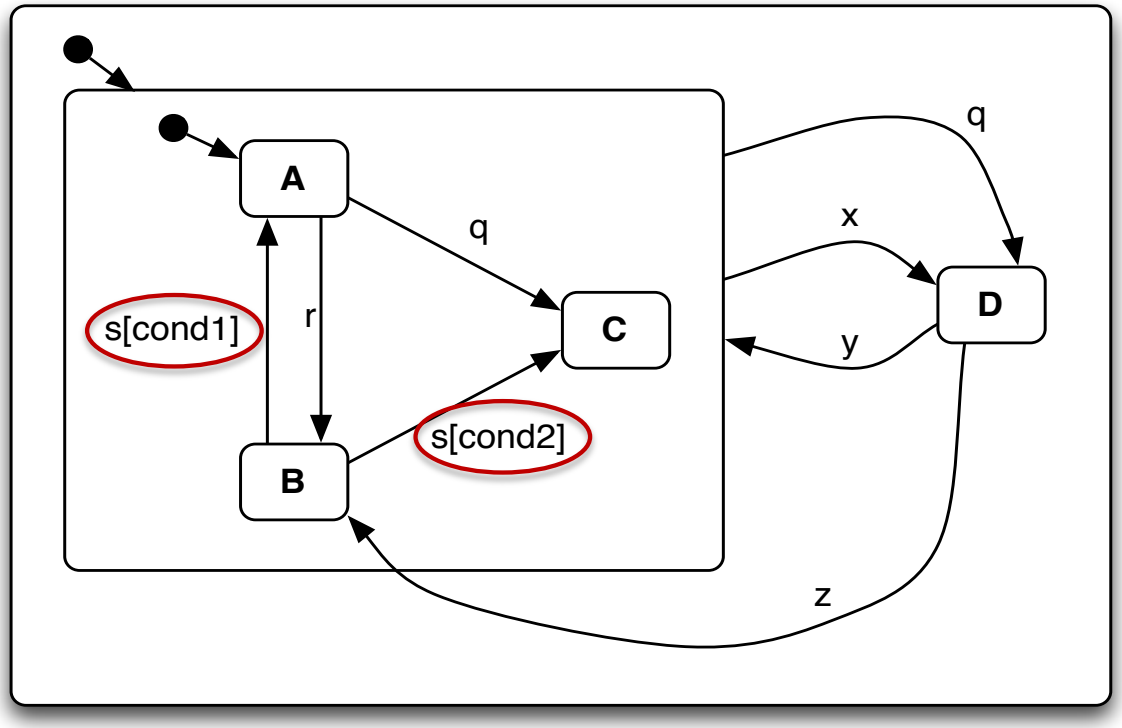

B • if event z occurs when in state D, what is the next state of the machine?

A • if event y occurs when in state D, what is the next state of the machine?

x • can the execution ever leave state C?

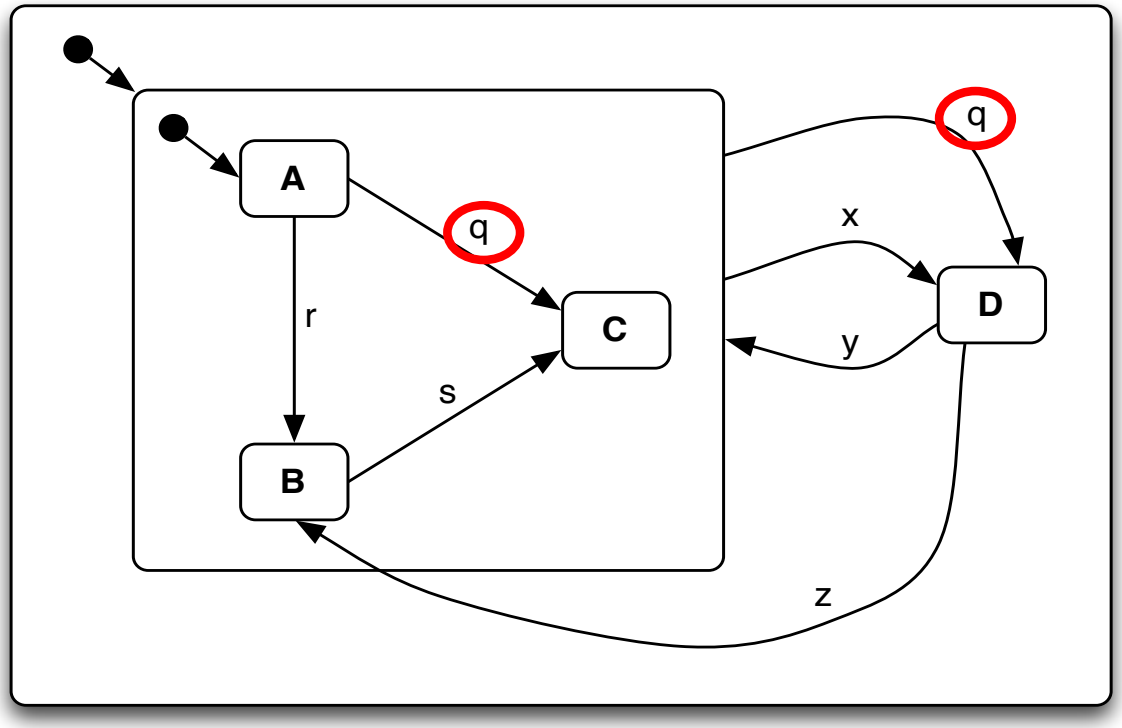
# Determinism

What if the machine is in state B and event s occurs?



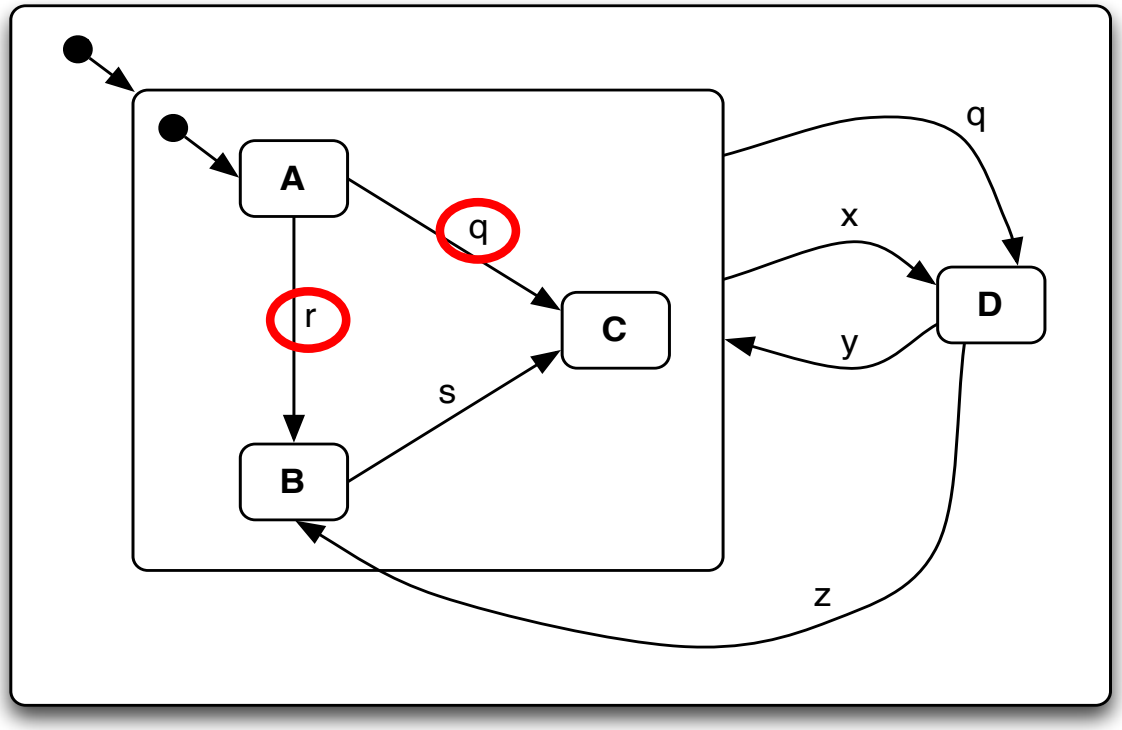
# Priority

What if the machine is in state A and event q occurs?



# Priority

What if the machine is in state A and events q and r occur simultaneously?

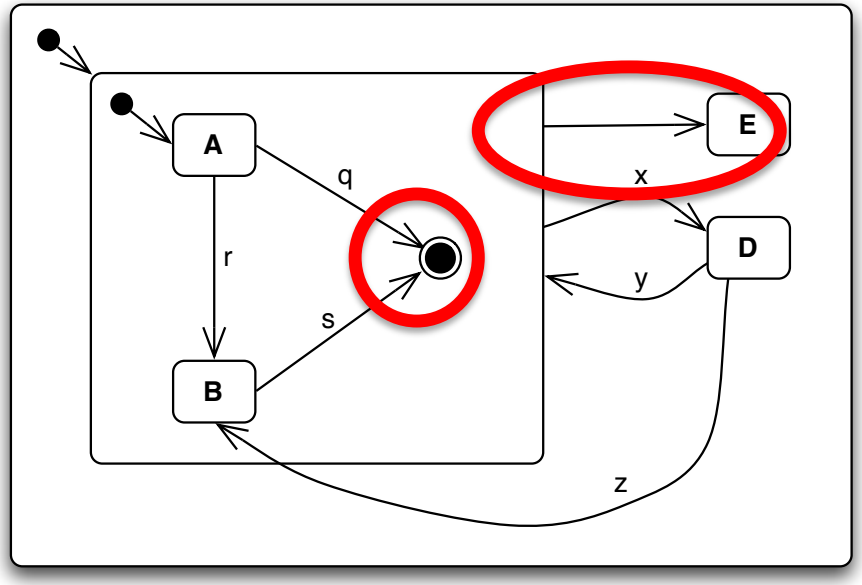


# Final state

A final state represents the end of computation within a superstate.

A transition that has no event or condition in its label is enabled when its source state is idle.

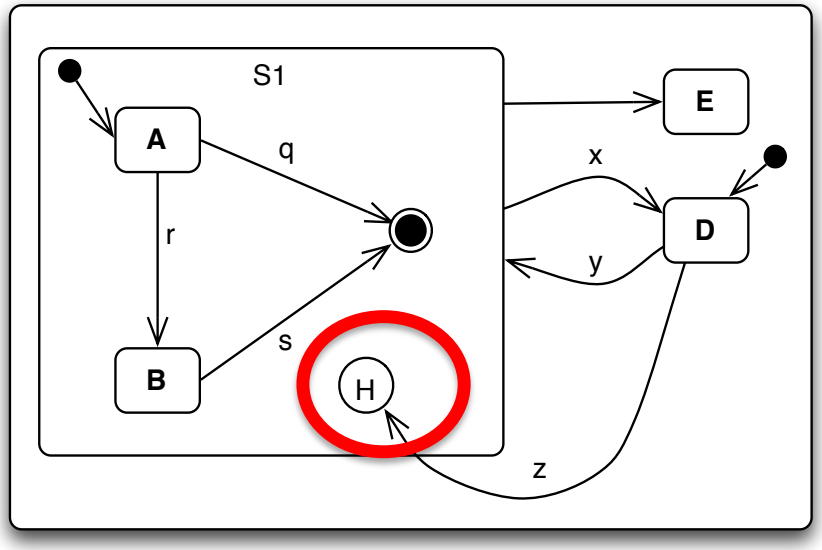
- source state is a basic state
- source superstate entered its final state
- source basic state has finished internal activity



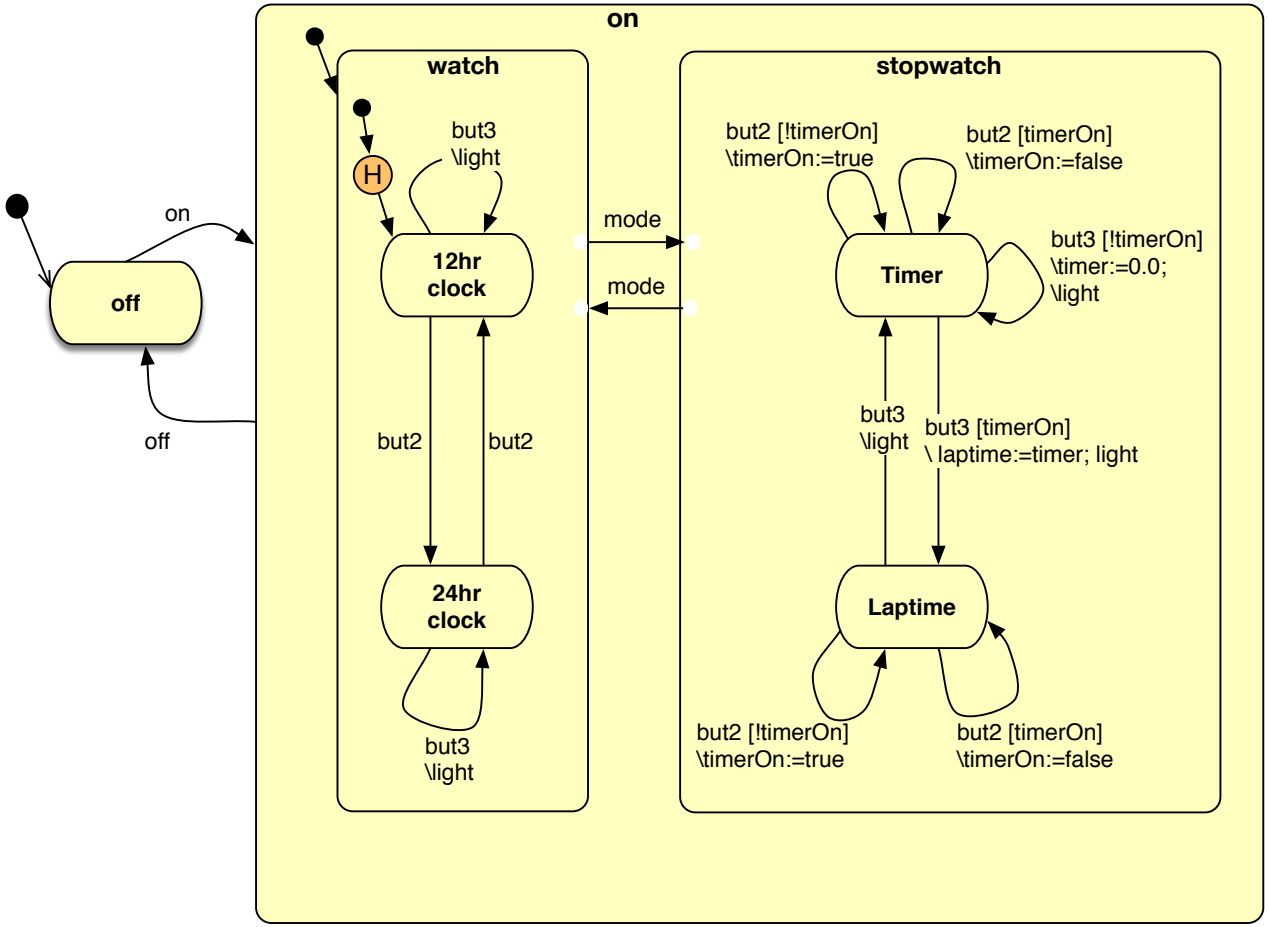

- If in A, what is the state after inputs rx? D
- If in A, what is the state after inputs qx? E

# History

History (H) is a pseudo-state that designates the child state of (H)'s parent state (i.e., child state of S1) that the execution was in when the parent state (S1) was last exited.



# History

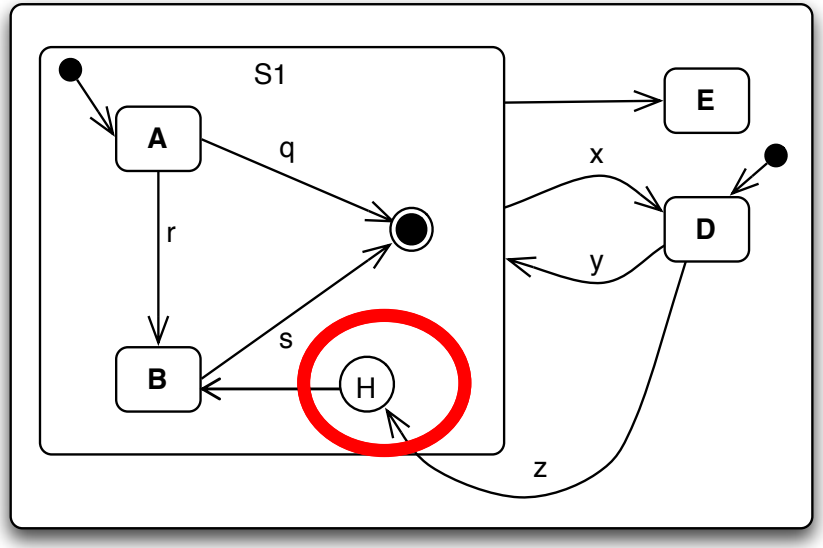


# History

History (H) is a pseudo-state that designates the child state of (H)'s parent state (i.e., child state of S1) that the execution was in when the parent state (S1) was last exited.

Cases:

- Superstate has history
  - last child state that was executing
- Superstate has no history
  - state designated by transition exiting history pseudo-state
  - superstate's default initial state

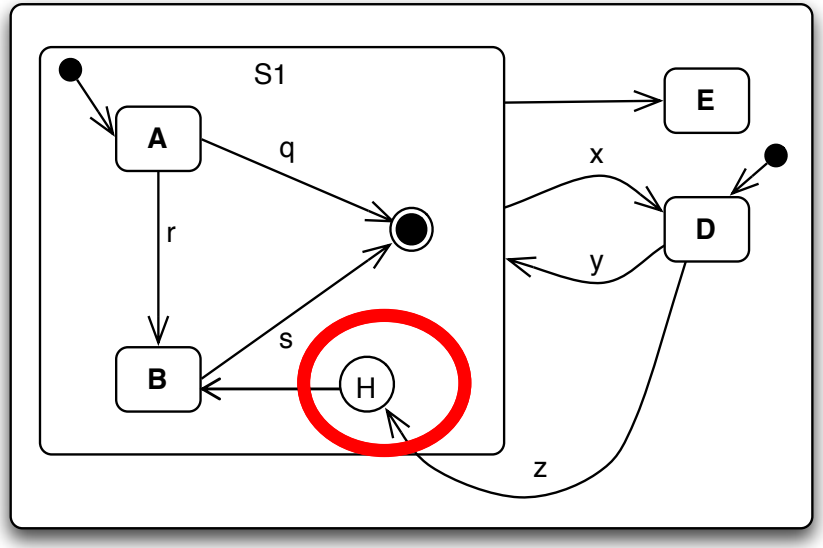


# History

History (H) is a pseudo-state that designates the child state of (H)'s parent state (i.e., child state of S1) that the execution was in when the parent state (S1) was last exited.

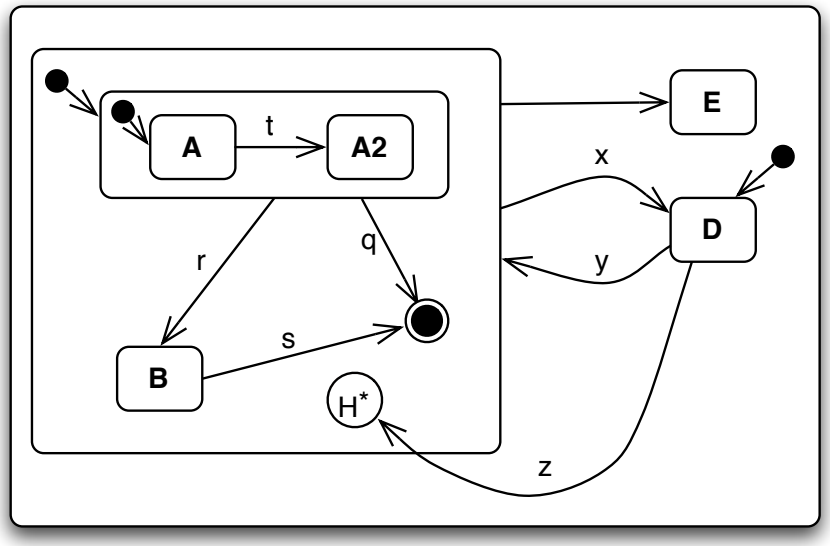
What is the current state after the following input sequences?

- y A
- yq E
- yrx D
- yrxz B
- z B



# Deep history: H*

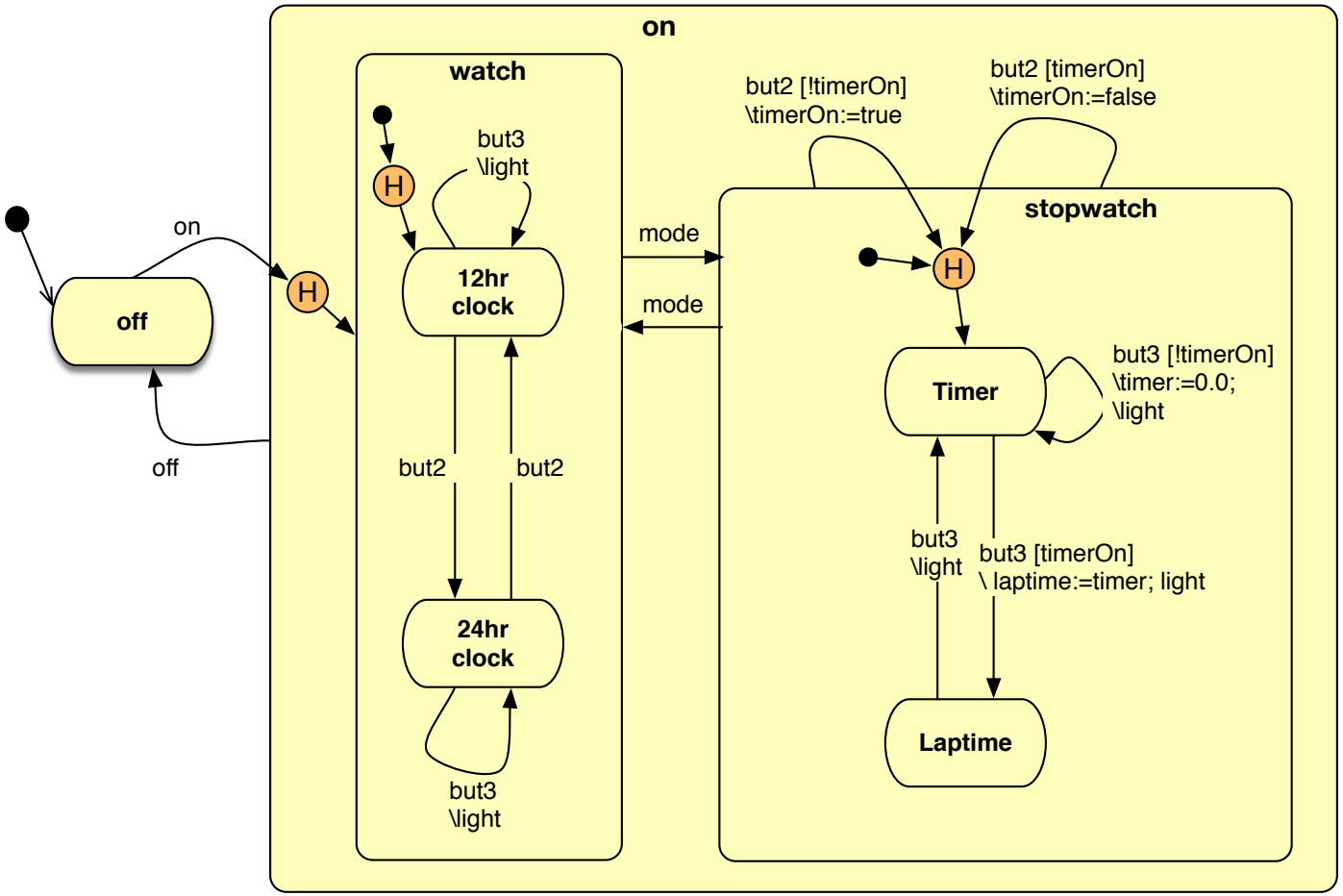
Deep history H* is a pseudo-state that designates the descendant state of H*'s parent state (i.e., descendent state of S1) that the execution was in the when the parent state (S1) was last existed.




What sequence of input events produces a different result if this model has a basic history vs. a deep history pseudo-state?

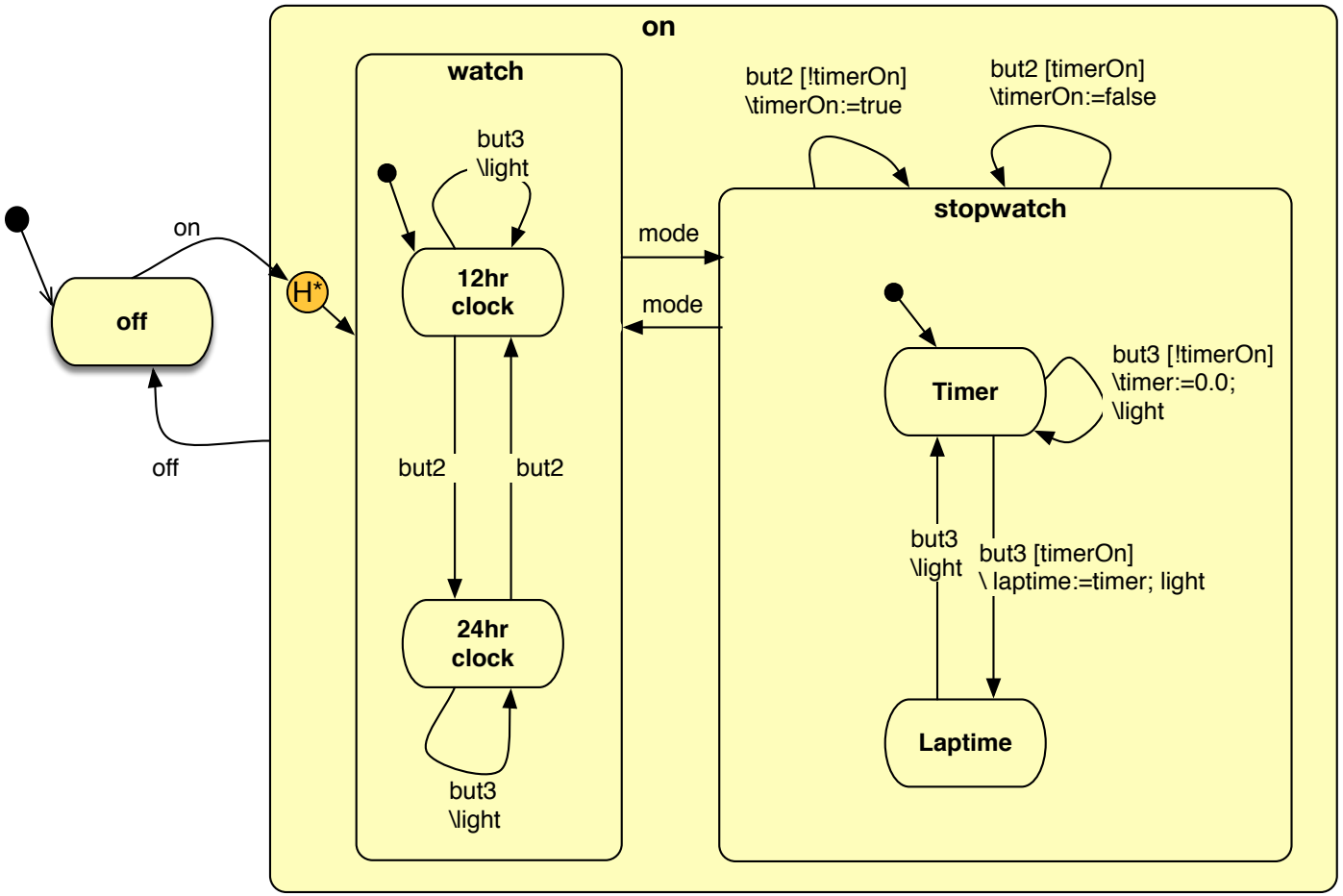
y t x z

# Stopwatch Example



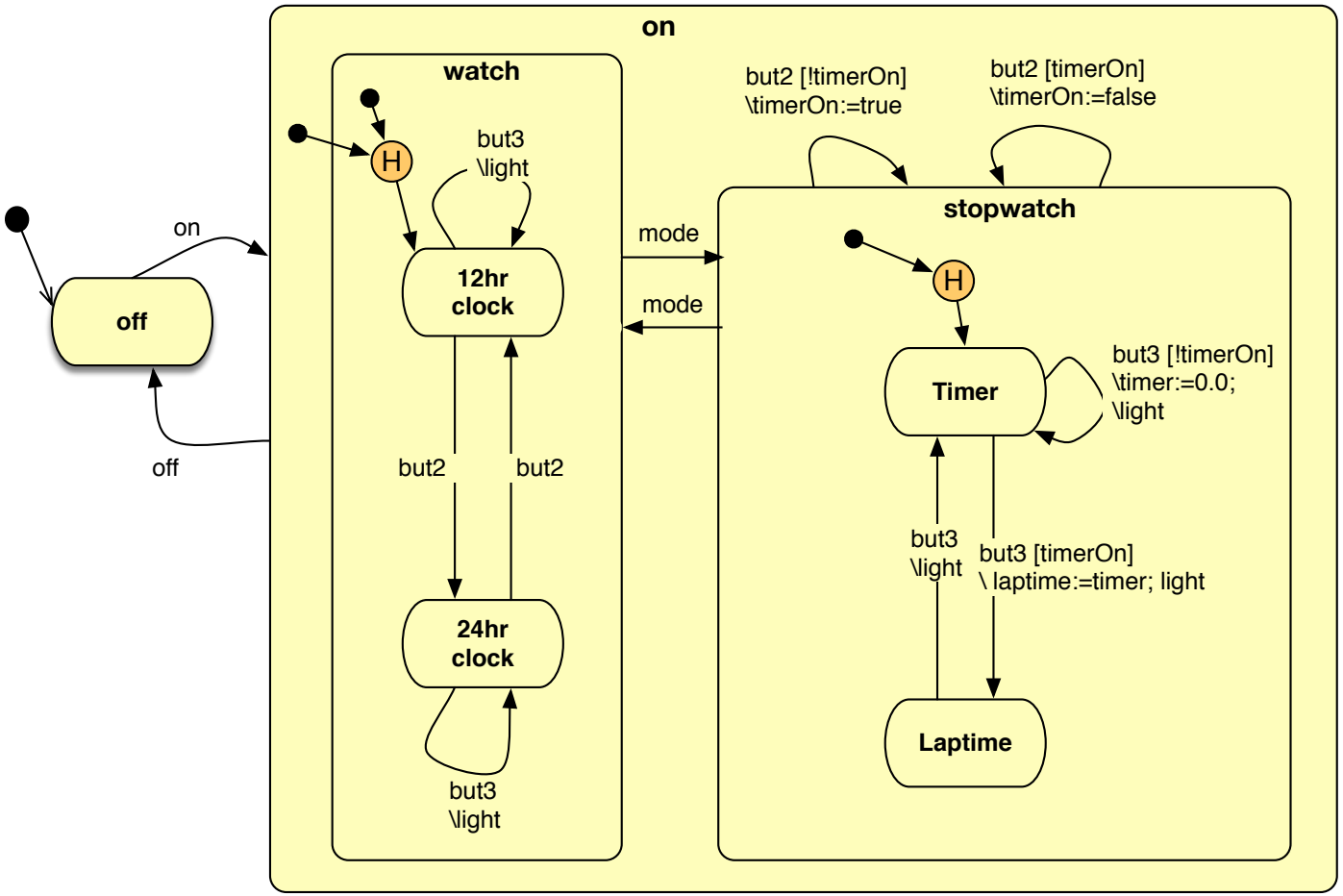

What state is the watch in when it is turned on?

# Stopwatch Example



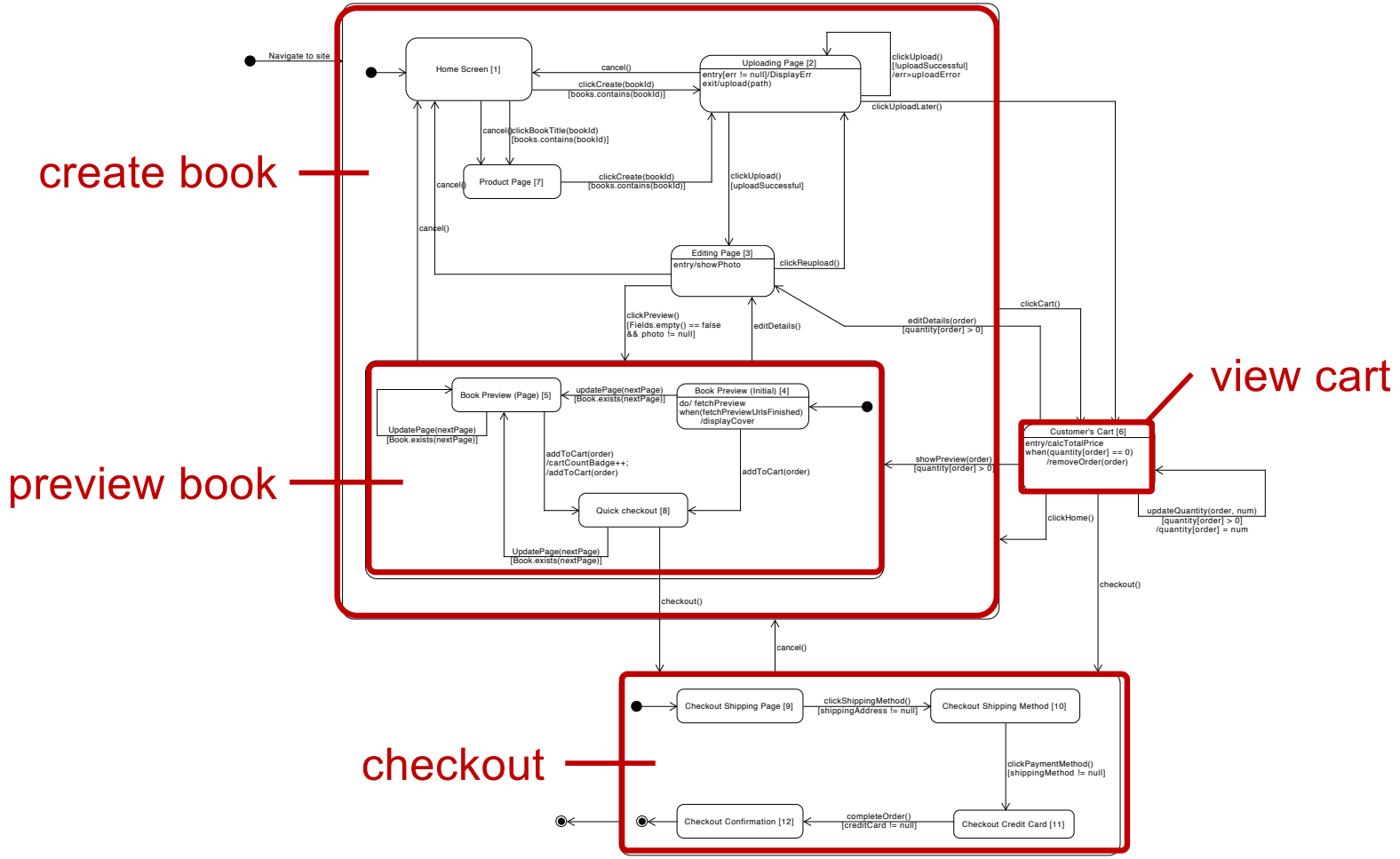

Is this model equivalent to the previous model (do they represent the same set of execution traces)?

# Stopwatch Example




What state is the watch in when it is turned on?

# Example: ReadYourStory



# References

Craig Larman, *Applying UML and Patterns, 3ed.,* Prentice Hall, 2004.

- Chapter 23: “UML State Machine Diagrams and Modeling”

Lenny Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*, Addison-Wesley Professional, 2013.

- Chapter 8: “State Machine Diagrams”