

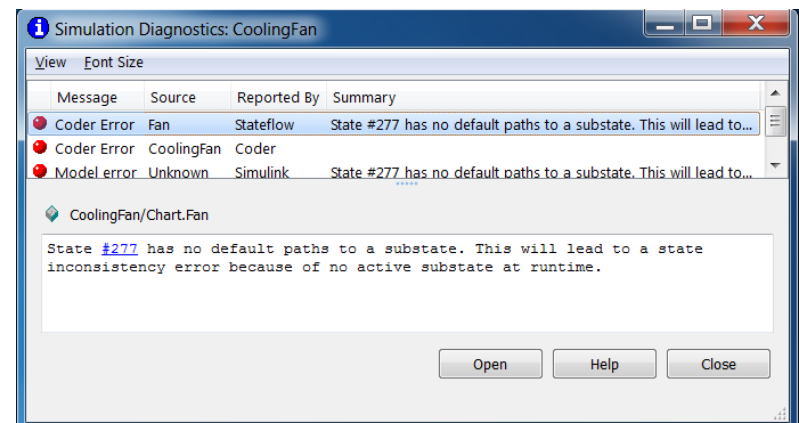
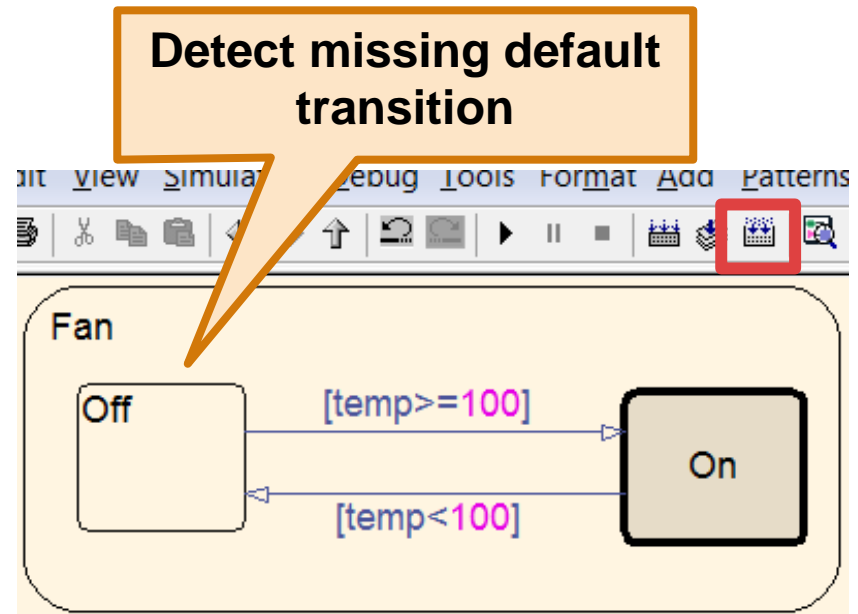
What's New in Stateflow

R2011b

Early Detection of State Inconsistencies

Identify state inconsistency errors before running the model

- New check to identify run-time state inconsistencies during model update
- Less dependence on testing to catch state inconsistencies

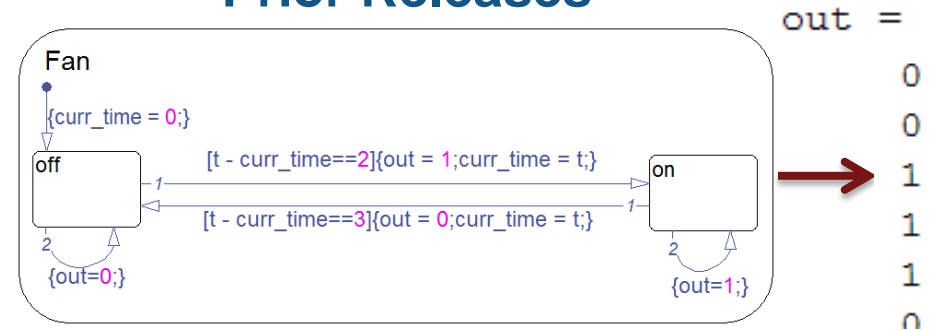


Persistent Outputs in Mealy and Moore Charts

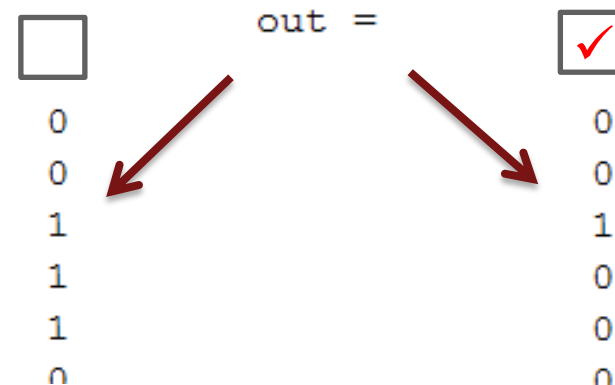
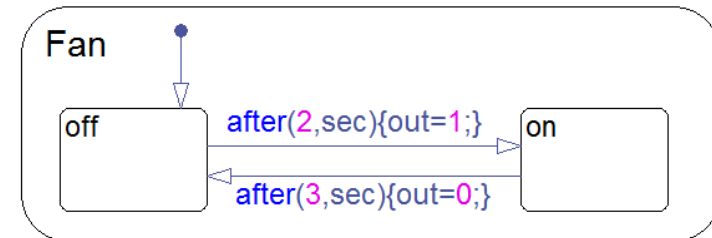
Create simpler Mealy and Moore charts

- Choose whether output variable values persist or reset each time chart wakes up
- Generate more efficient code

Prior Releases



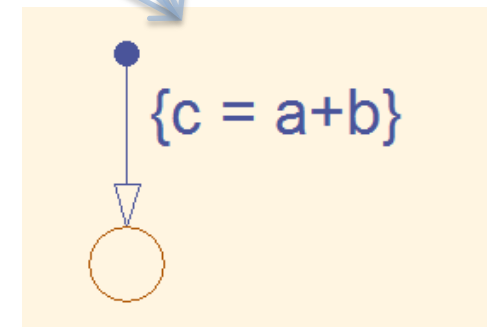
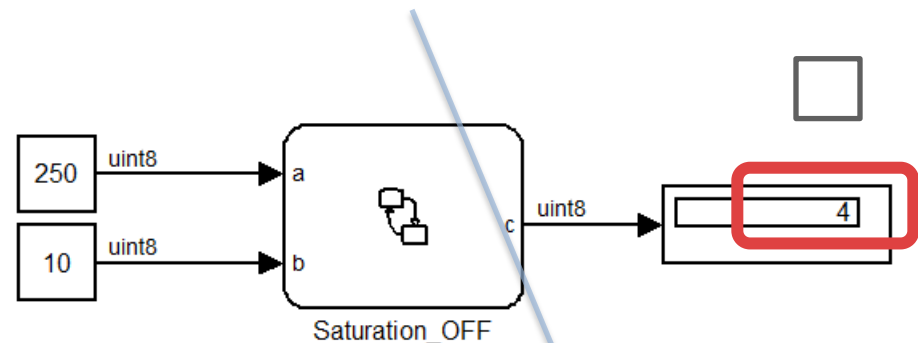
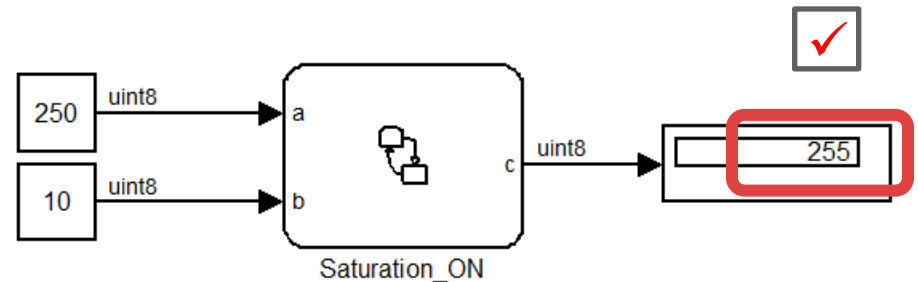
In R2011b



Saturation on Overflow

Generate robust code for high-integrity applications

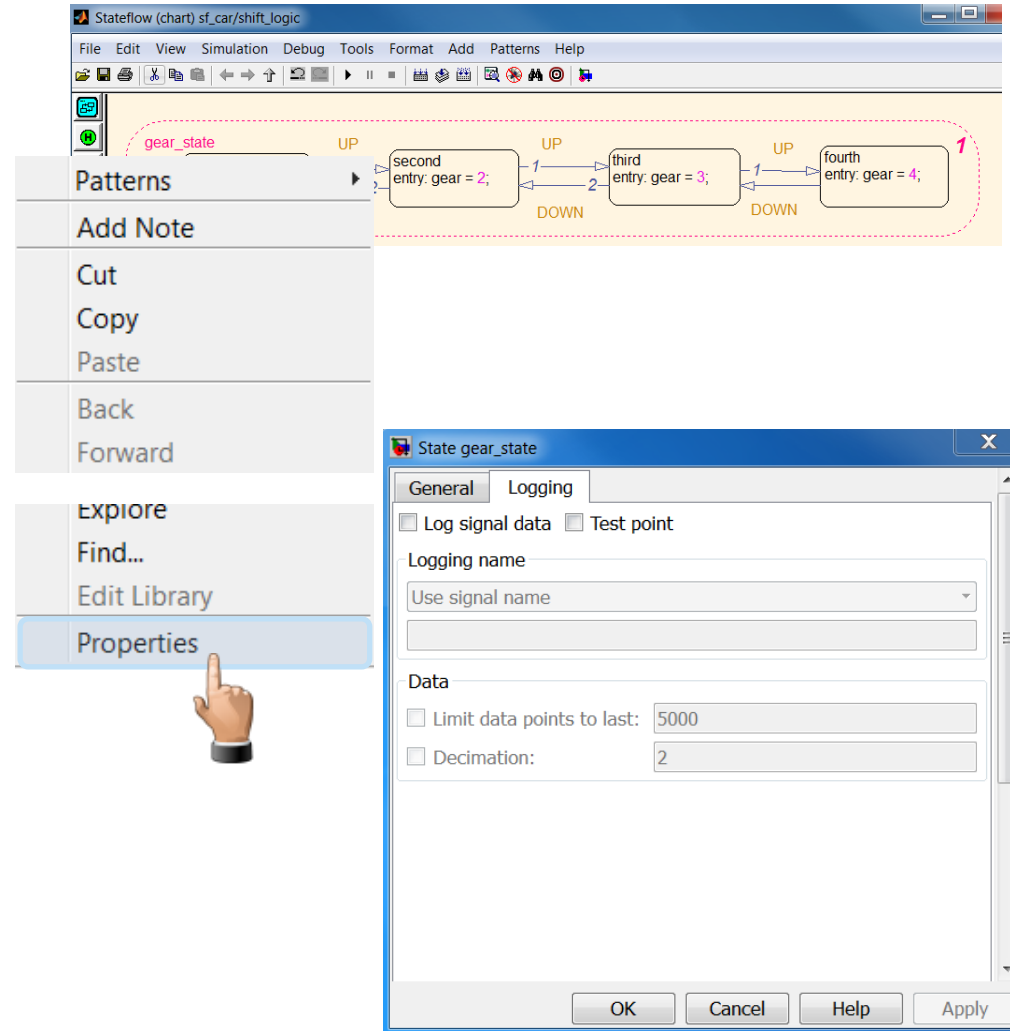
- Saturate integer data on overflow with new default chart option – for more robust code
- Wrap integer data on overflow by deselecting chart option – for more efficient code



New User Interface for Signal Logging

Easier setup of state and data logging configuration

- New Logging tab on state and data properties consistent with Simulink



The screenshot displays the Stateflow interface for a state machine diagram titled 'sf_car/shift_logic'. The diagram shows three states: 'second entry: gear = 2;', 'third entry: gear = 3;', and 'fourth entry: gear = 4;'. Transitions are labeled 'UP' and 'DOWN'. A context menu is open over the diagram, with the 'Properties' option highlighted by a hand cursor. The 'Properties' dialog box is open, showing the 'Logging' tab. The 'Logging' tab includes options for 'Log signal data' and 'Test point', a 'Logging name' field set to 'Use signal name', and a 'Data' section with 'Limit data points to last' set to 5000 and 'Decimation' set to 2. The dialog box has 'OK', 'Cancel', 'Help', and 'Apply' buttons at the bottom.

Default Case in Switch Statements



Improve code coverage or MISRA compliance with new code style option

- Control generation of default case in switch statements using new code style option

```
switch (sf_car_DWork.is_gear_state) {
    case sf_car_IN_first:

    }
    break;
    default:
        sf_car_DWork.is_gear_state = (uint8_T)sf_ca
        break;
}
}
```



```
switch (sf_car_DWork.is_gear_state) {
    case sf_car_IN_first:

    }
    break;
}
}
```

Portion of generated C code