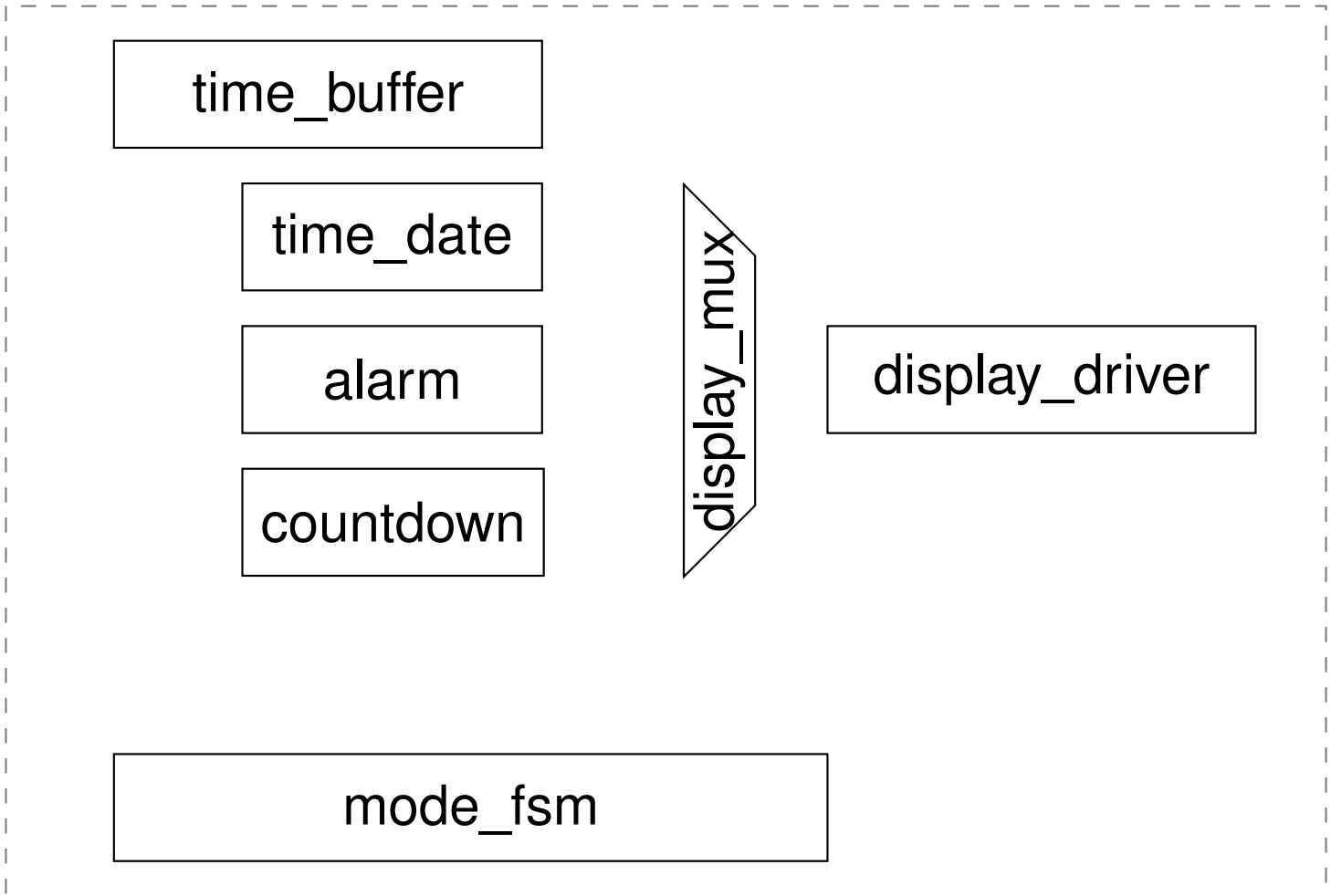
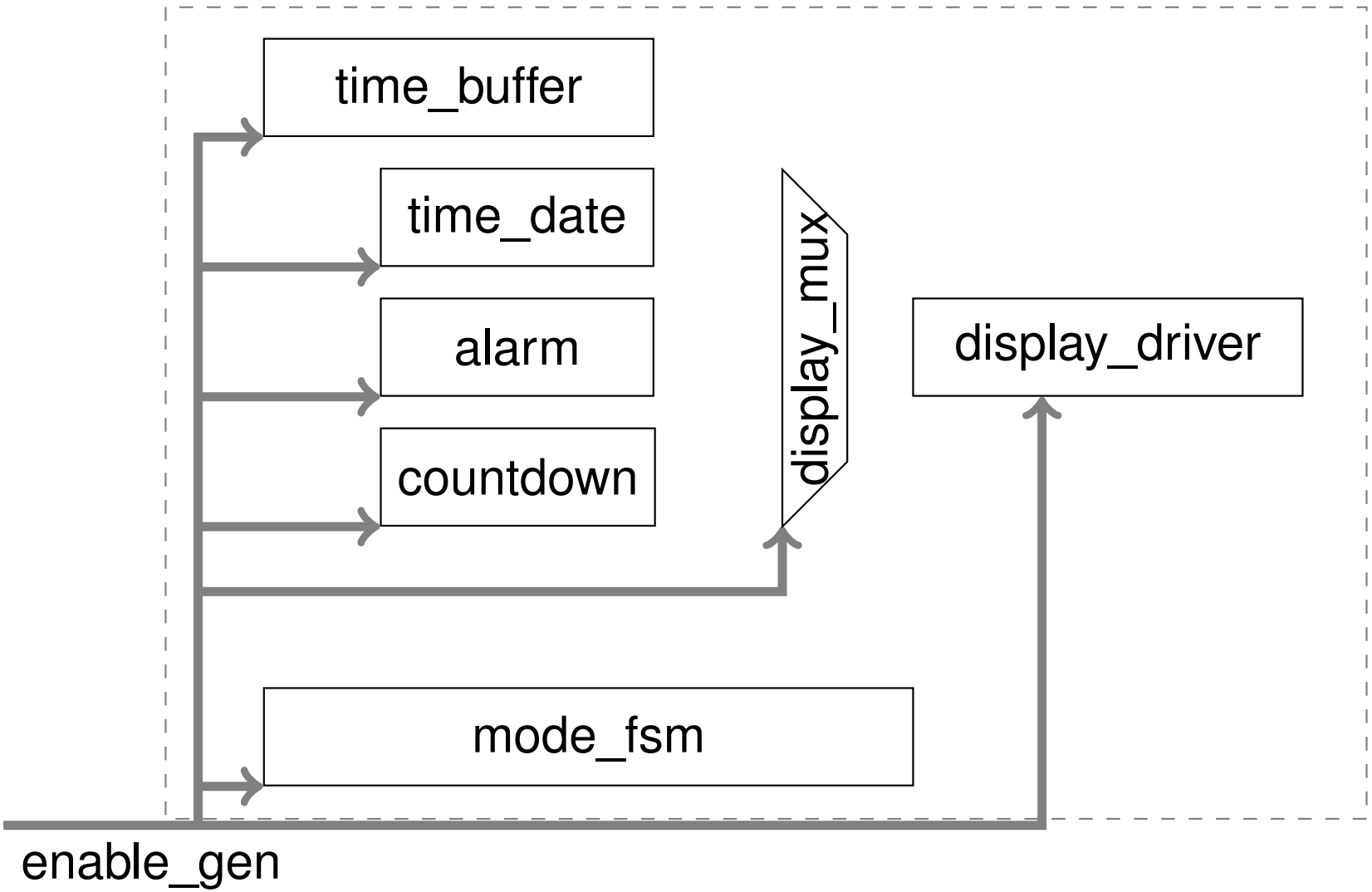
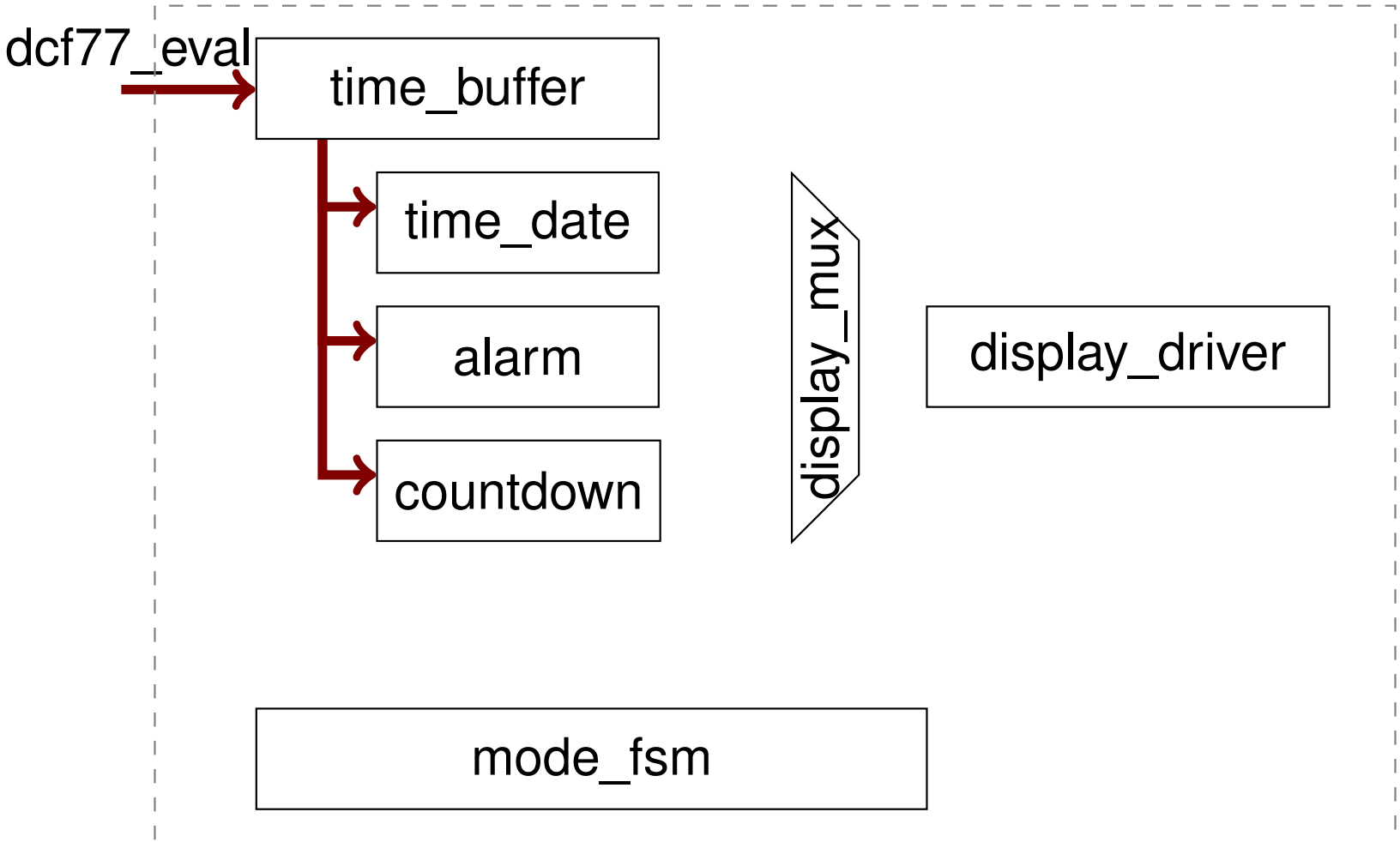
Partitioning of Clock Functionality



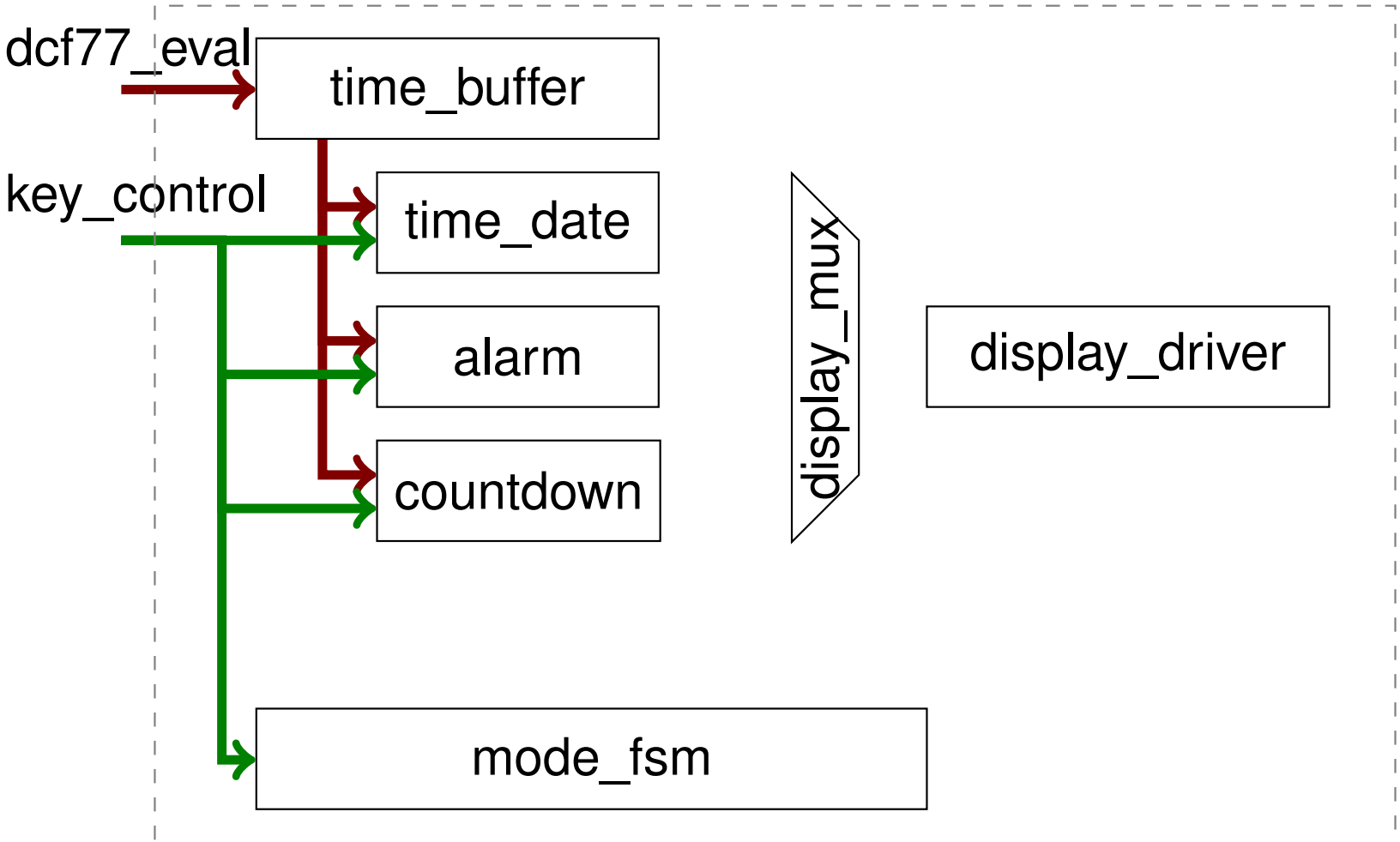
Partitioning of Clock Functionality



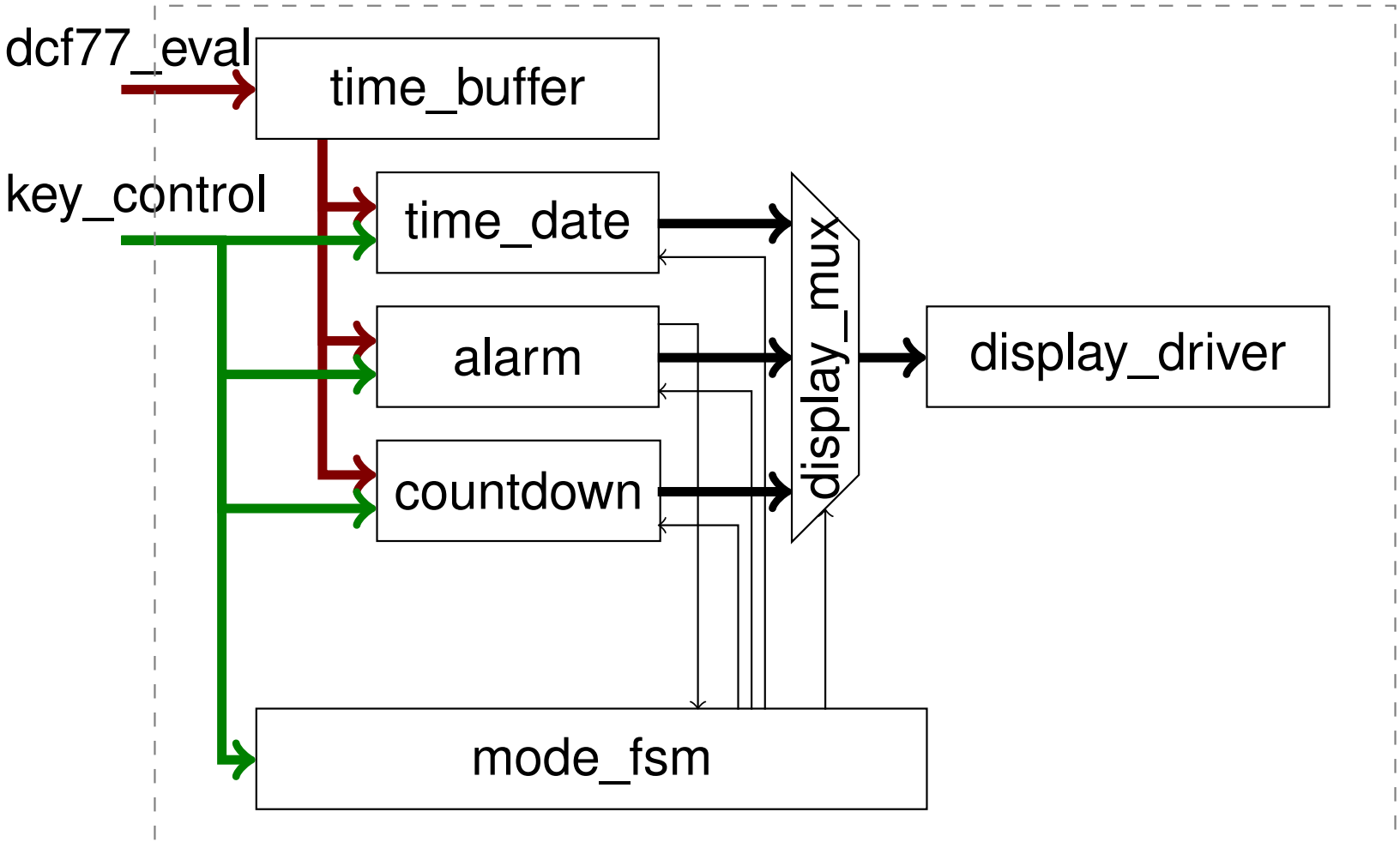
Partitioning of Clock Functionality



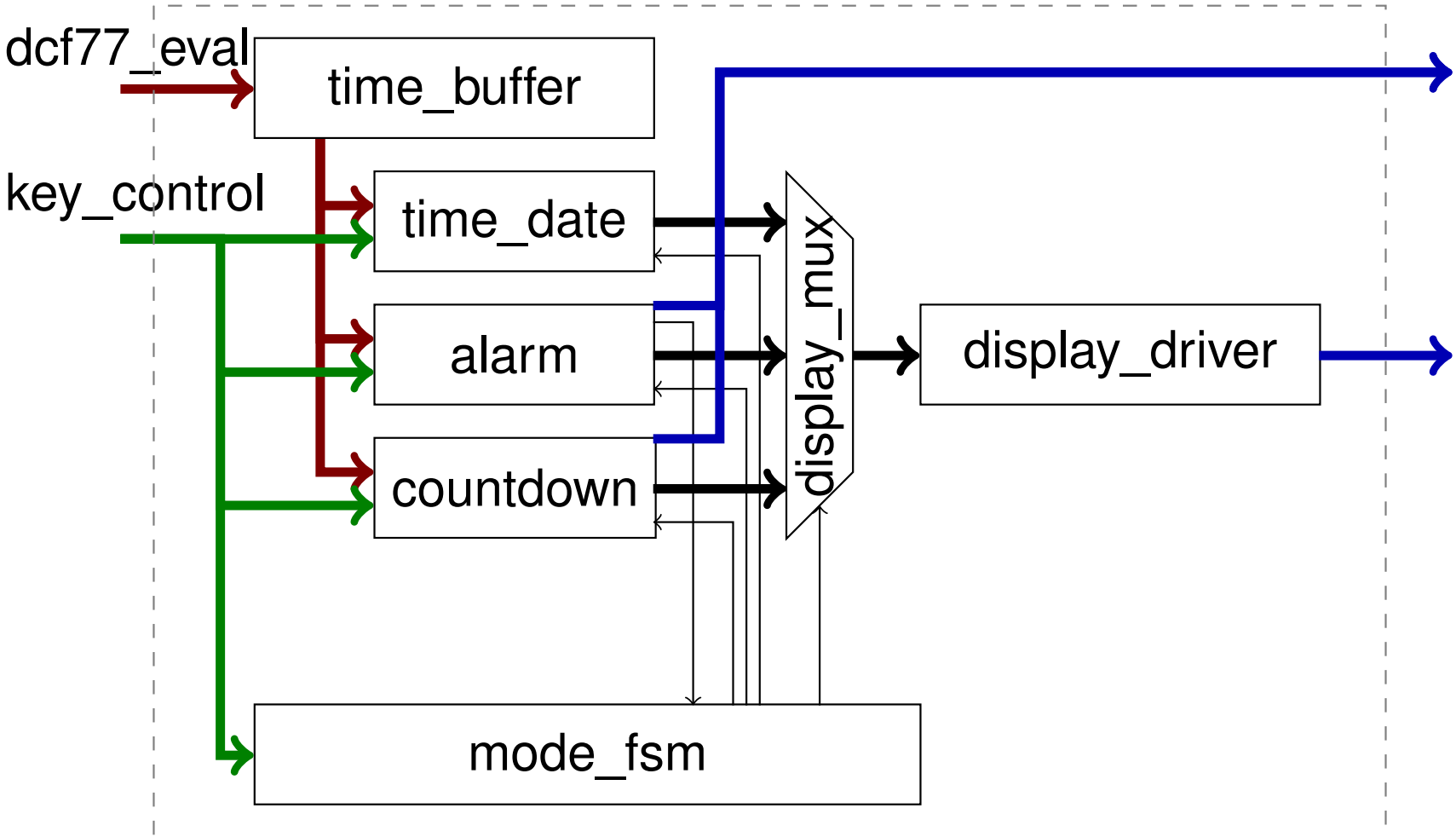
Partitioning of Clock Functionality



Partitioning of Clock Functionality

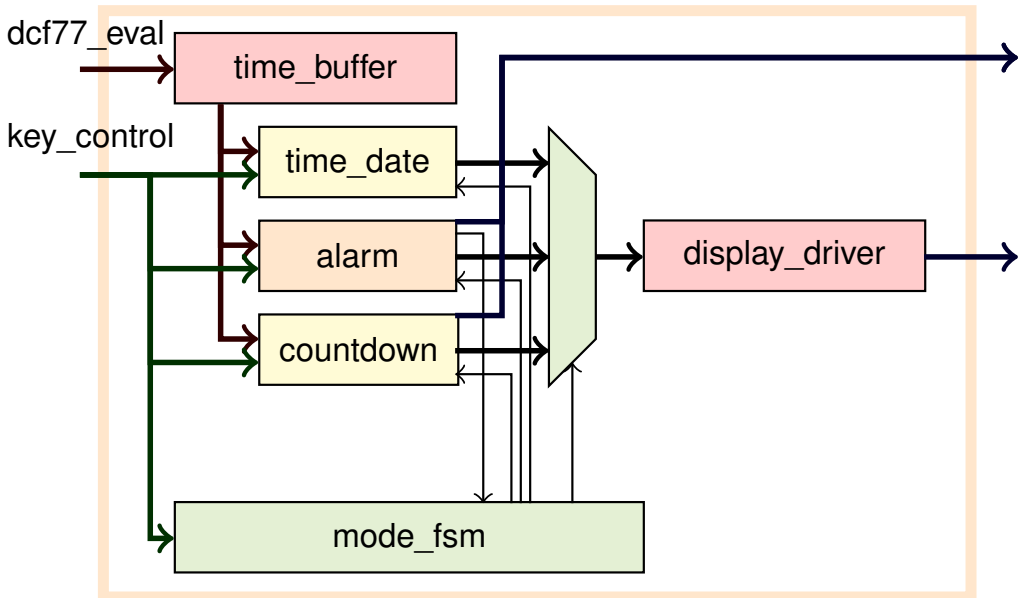


Partitioning of Clock Functionality



Workload Distribution

uhrenbaustein	Sophia
mode_alarm	Sophia
mode_fsm	Fabian
display_mux	Fabian
mode_time_date	Tobi
mode_countdown	Tobi
time_buffer	Mathias
display_driver	Mathias



Common System Integration

- Common data format: VHDL package for all modules
- Use git to synchronize efforts
- E-Mail for discussion
- Top level module is treated like any other module

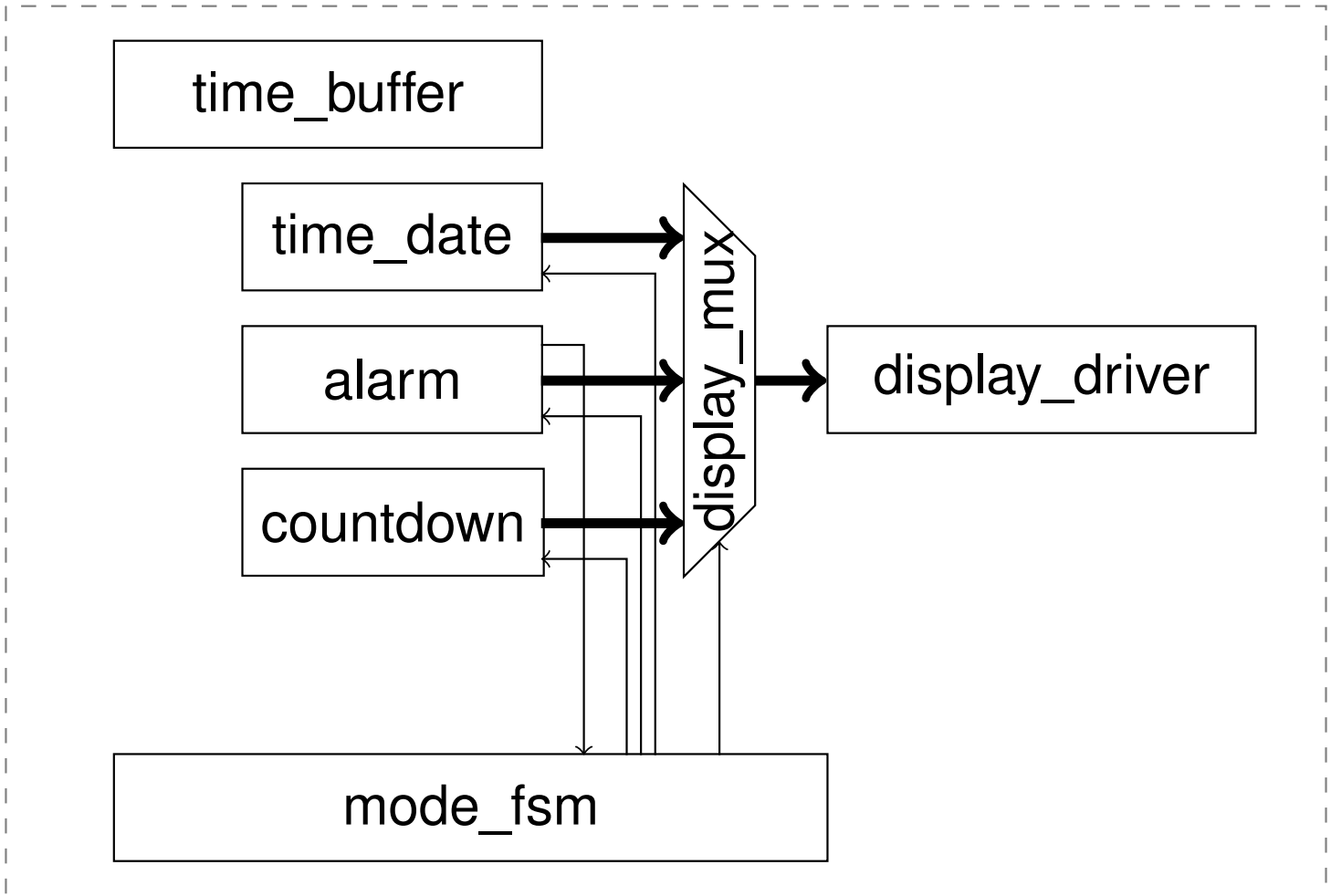
Testing strategy

1. person assigned to module creates individual testbench
2. test corner cases, e.g.:
   - correct behavior when alarm is ringing
   - invalid DCF77 signal
   - reset
   - continued countdown in background
3. supervisor checks module and testbench after realization
4. tests on actual hardware

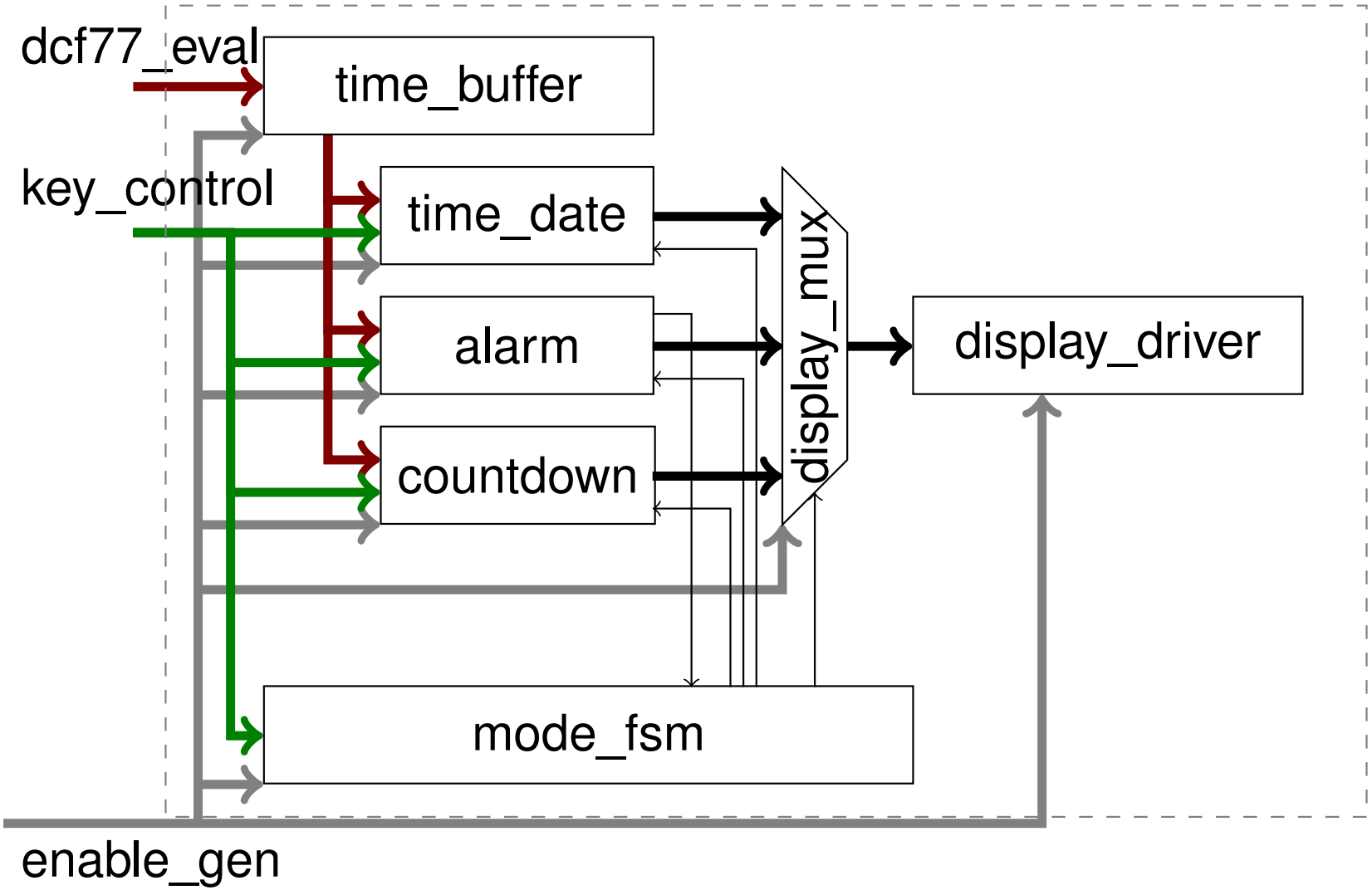
Testing strategy

	implement	supervise
uhrenbaustein	Sophia	Fabian
mode_alarm	Sophia	Fabian
mode_fsm	Fabian	Tobi
display_mux	Fabian	Tobi
mode_time_date	Tobi	Mathias
mode_countdown	Tobi	Mathias
display_driver	Mathias	Sophia
time_buffer	Mathias	Sophia

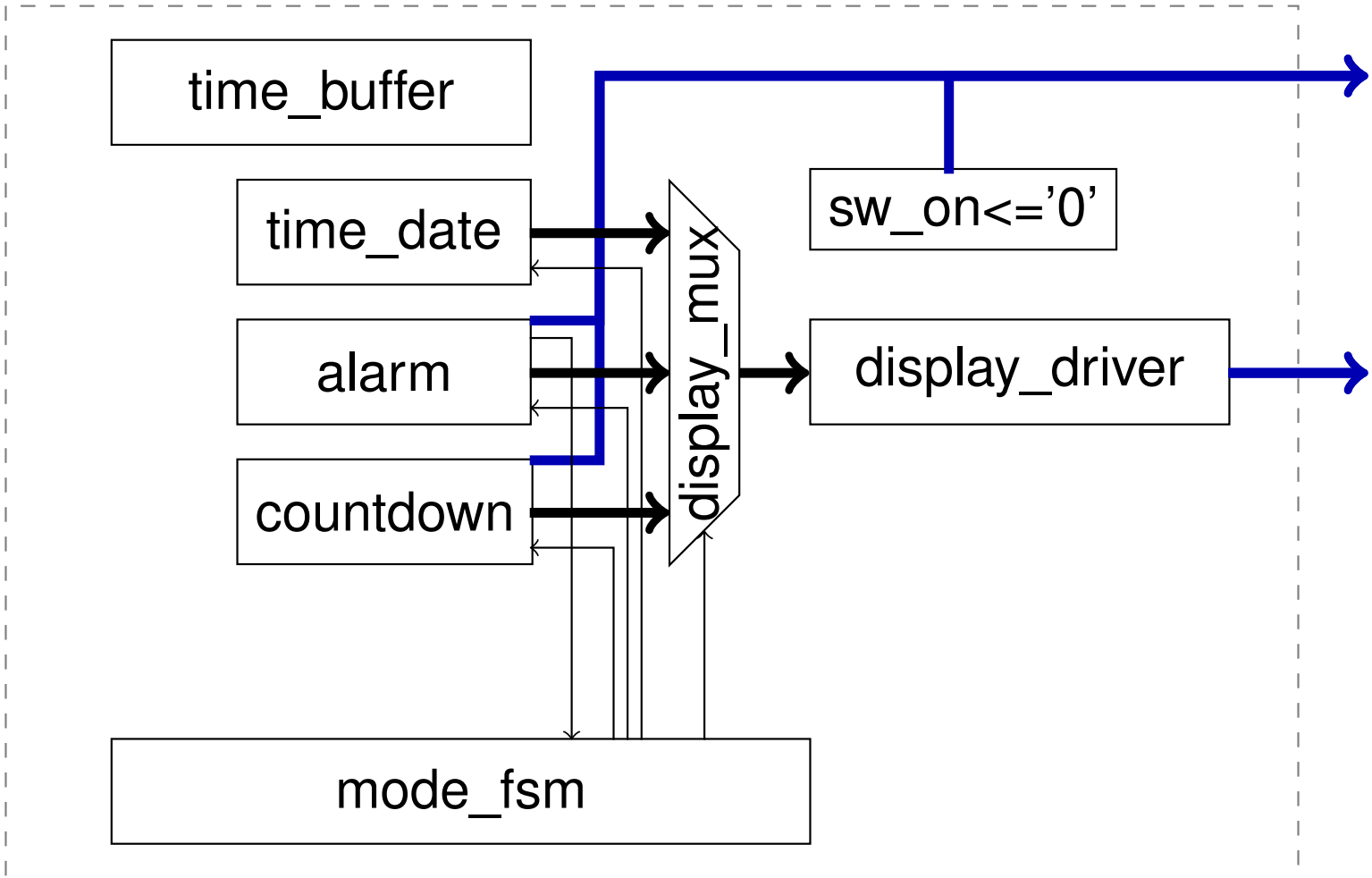
uhrenbaustein - integrate modules



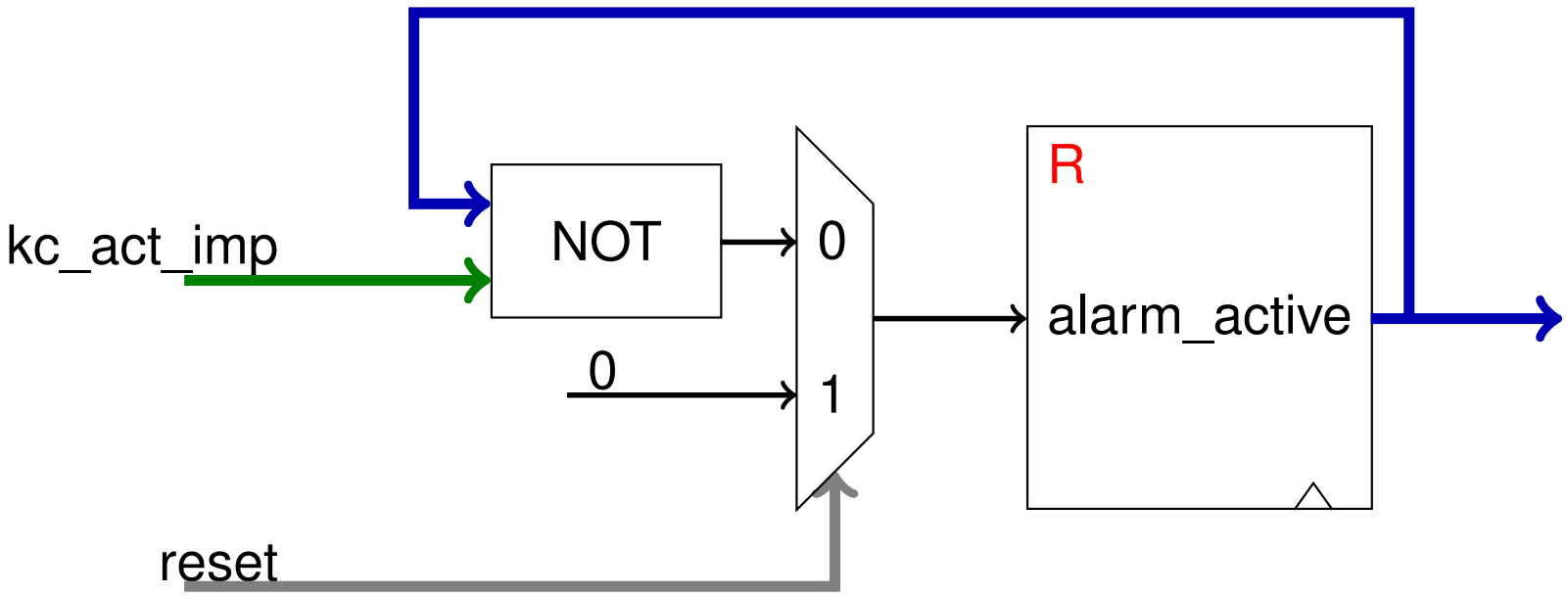
uhrenbaustein - supply input signals



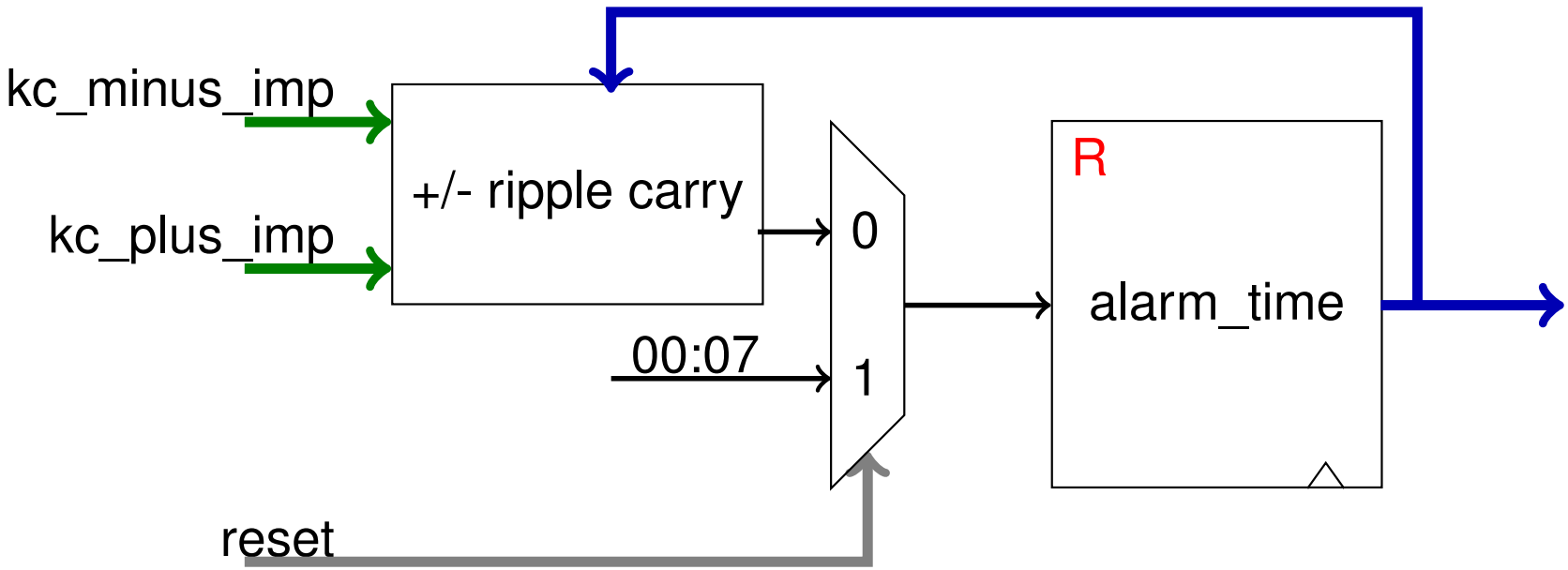
uhrenbaustein - supply output signals



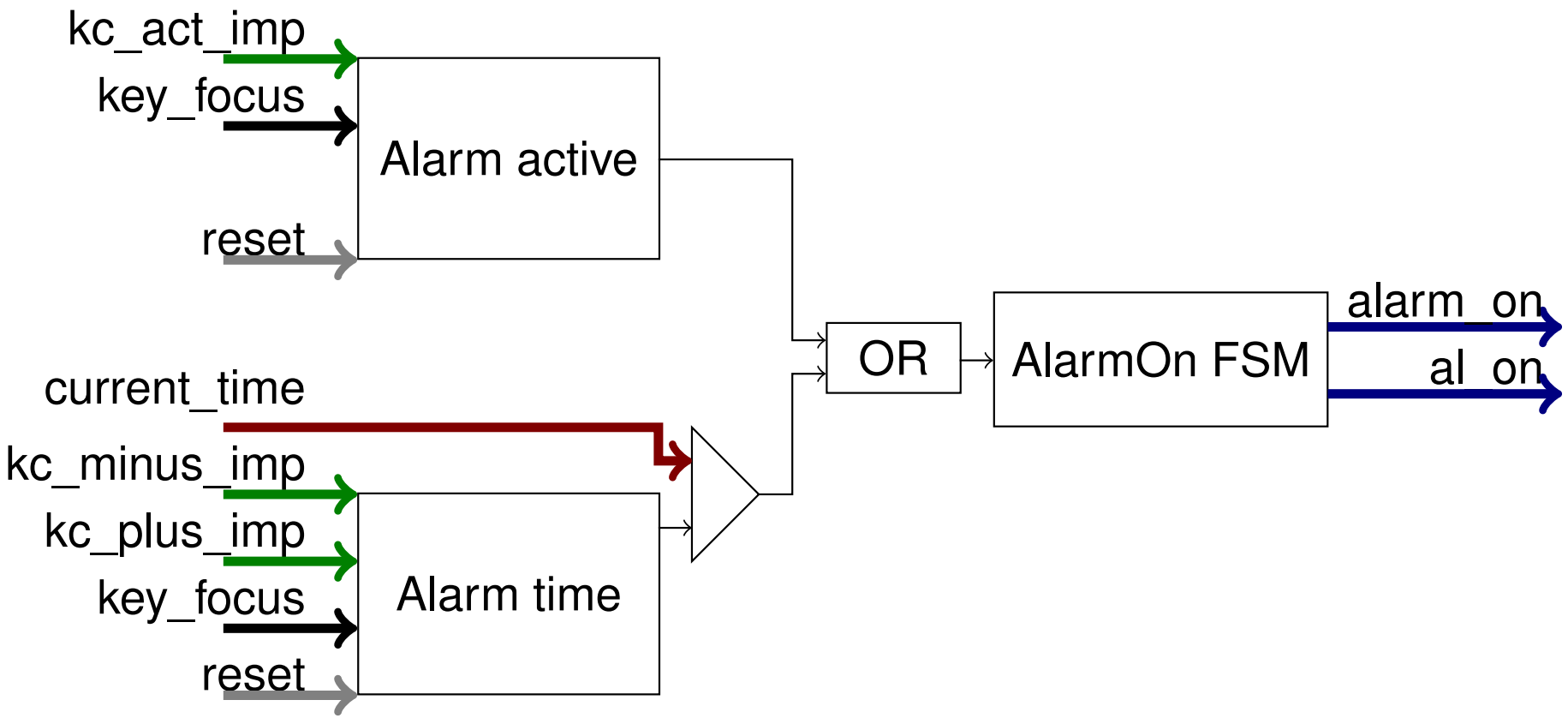
Alarm active



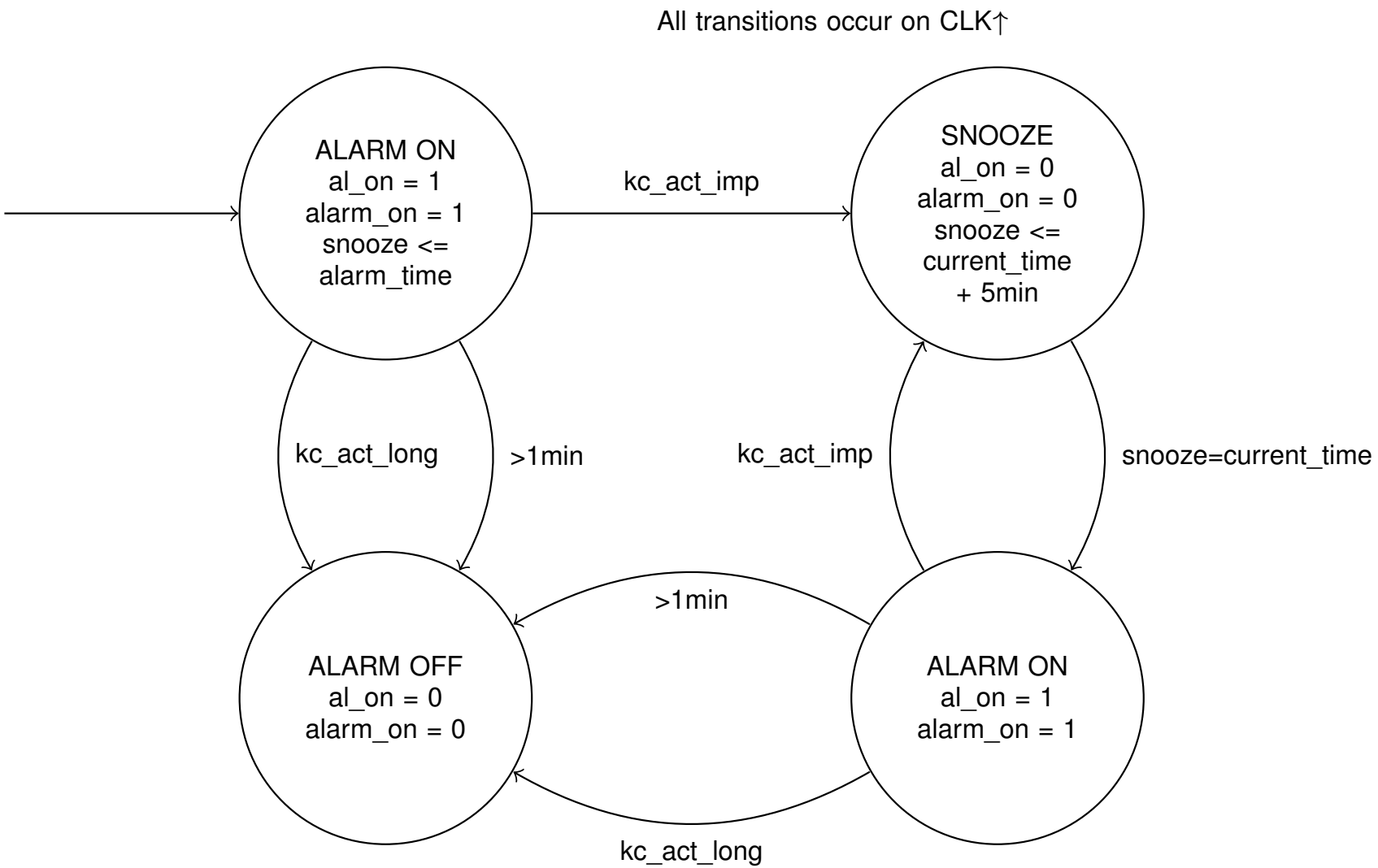
Alarm time


kc_minus_imp
kc_plus_imp
+/- ripple carry
0
00:07
1
R
alarm_time
reset

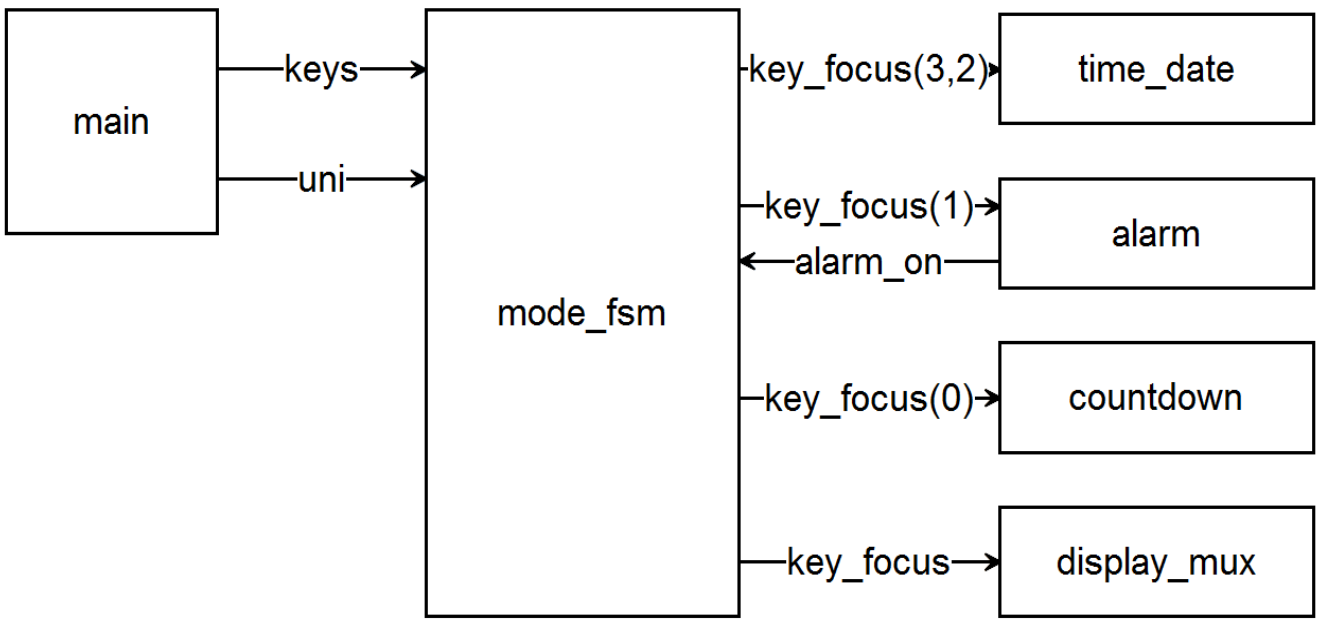
mode_alarm overview



mode_alarm FSM:AlarmOn



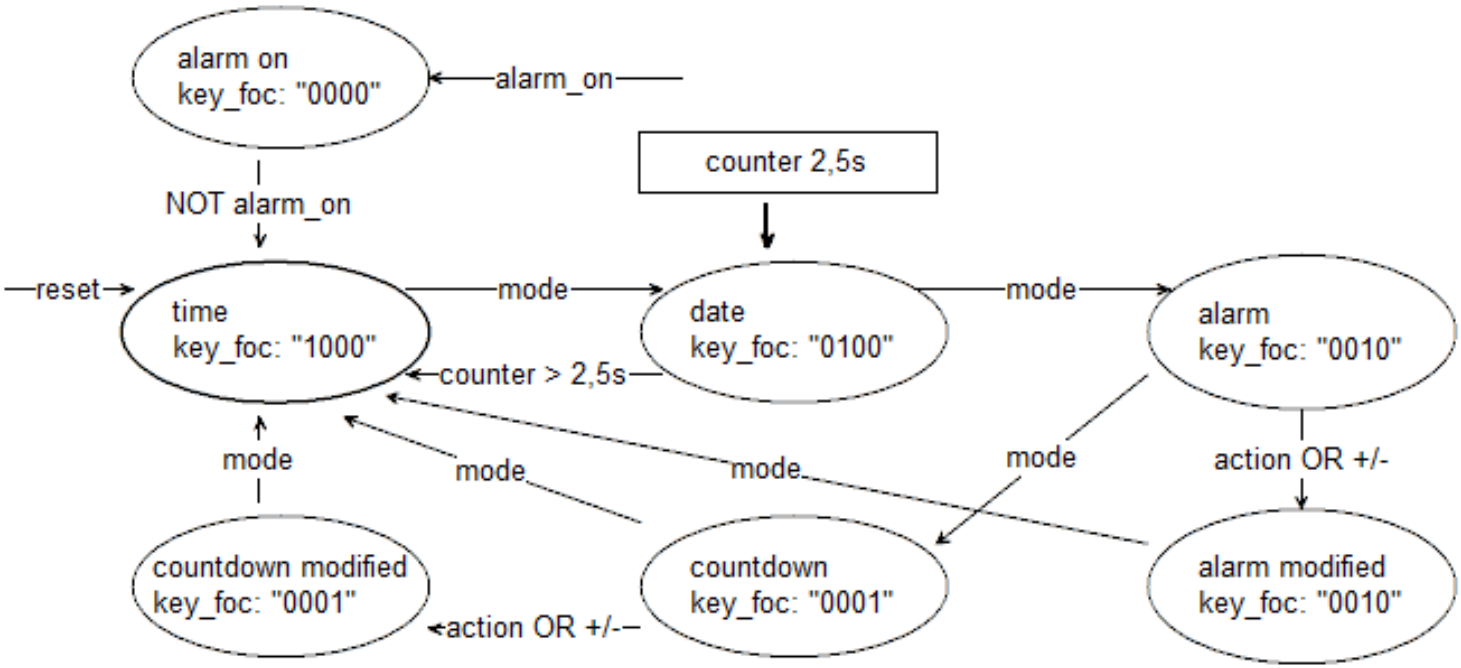
mode FSM Interface



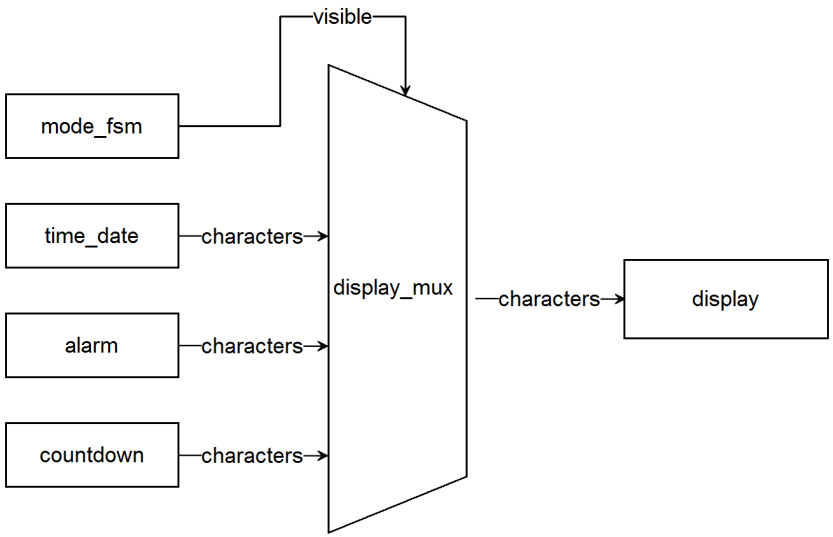
Mode _fsm Funktion

- Controls current mode
- Handles key access of modules (special case alarm ringing)
- Creates control signal for display_mux

Mode _fsm state diagram



Display MUX Interface



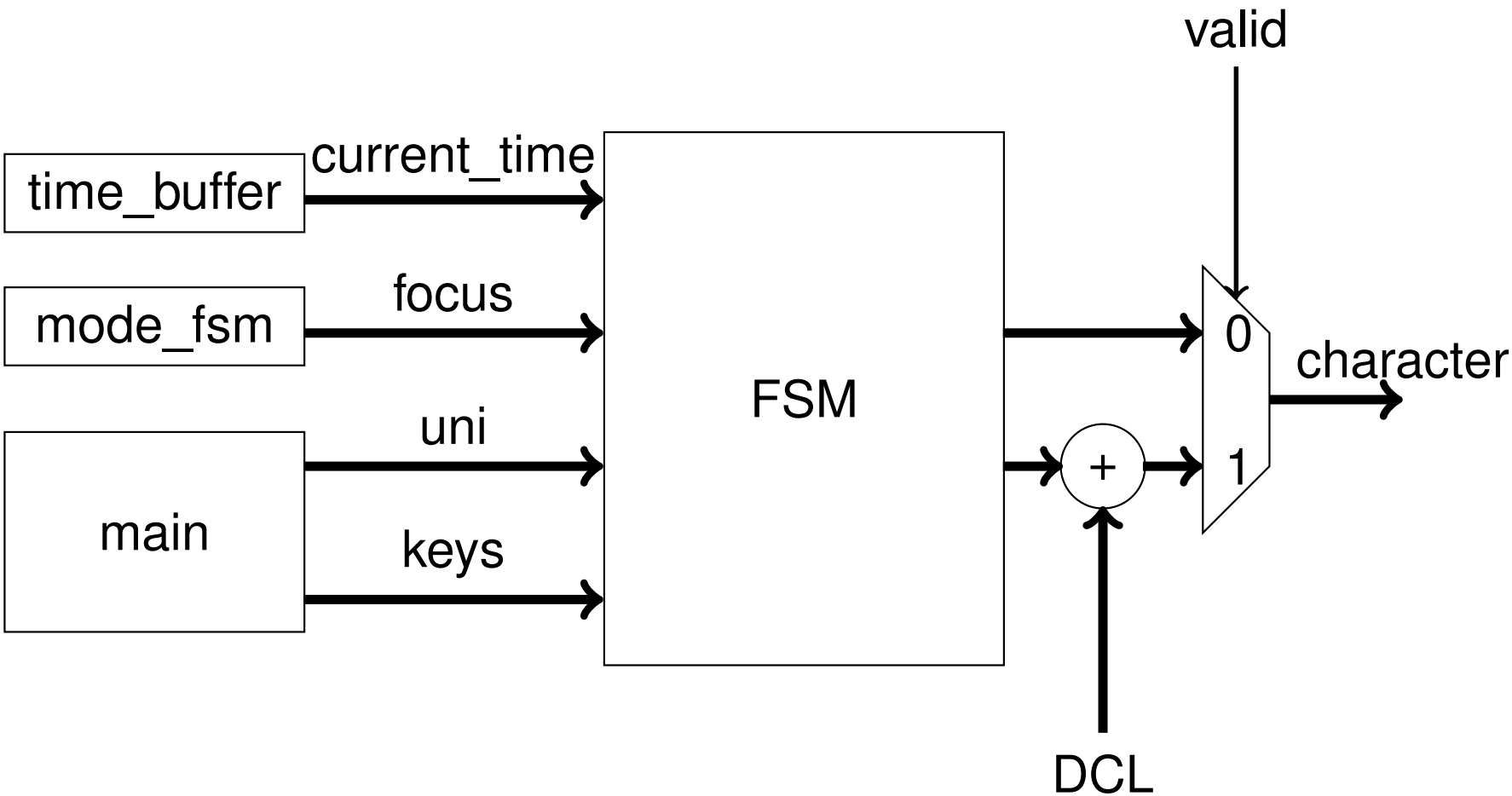
- forwards characters of active module
- "DCF" string and star for active alarm always from time_date/alarm module

Time and Date component

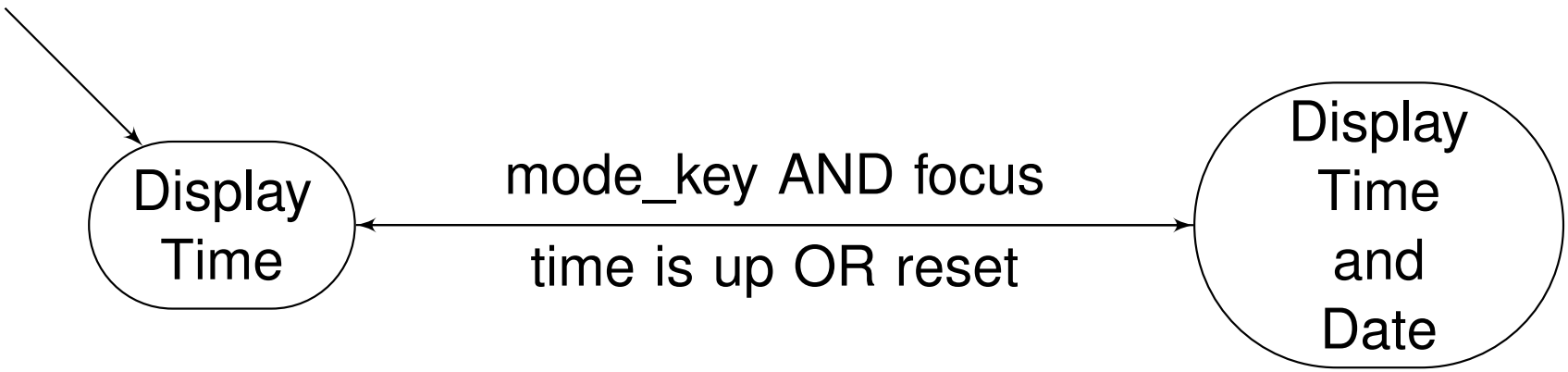
```
component time_date is
 port(
  current_time:    in  time_signals;
  keyboard_focus: in  std_logic_vector(2 downto 0);
  uni:             in  universal_signals;
  keys:            in  keypad_signals;
  valid:           in  std_logic;

  character:      out character_array_2d(3 downto 0,
                                         19 downto 0)
 );
end component;
```

Time and Date Overview



Time and Date FSM

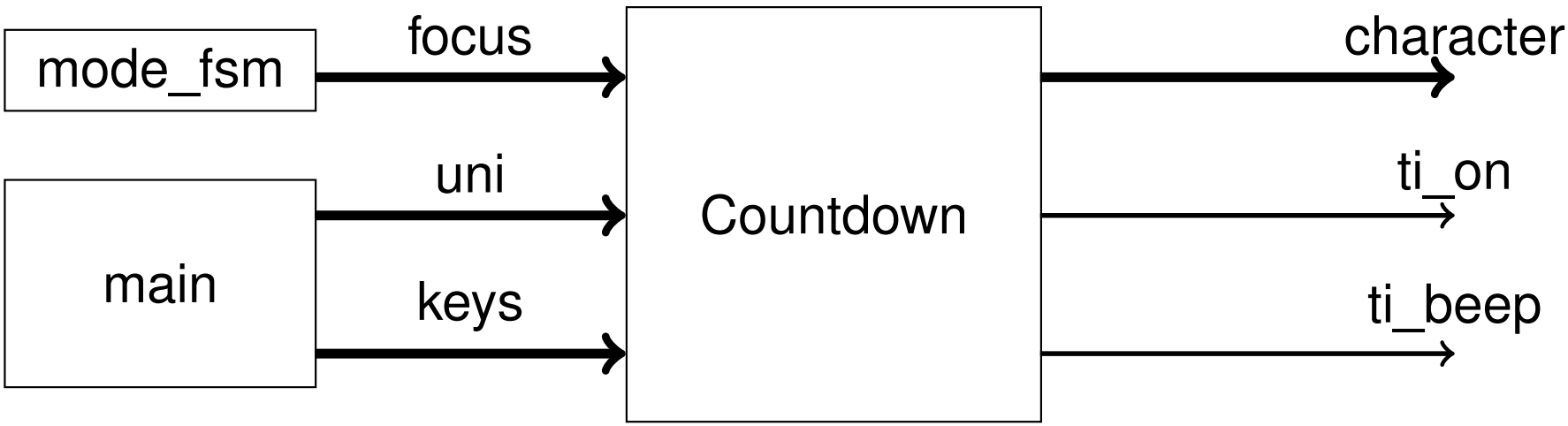


Countdown component

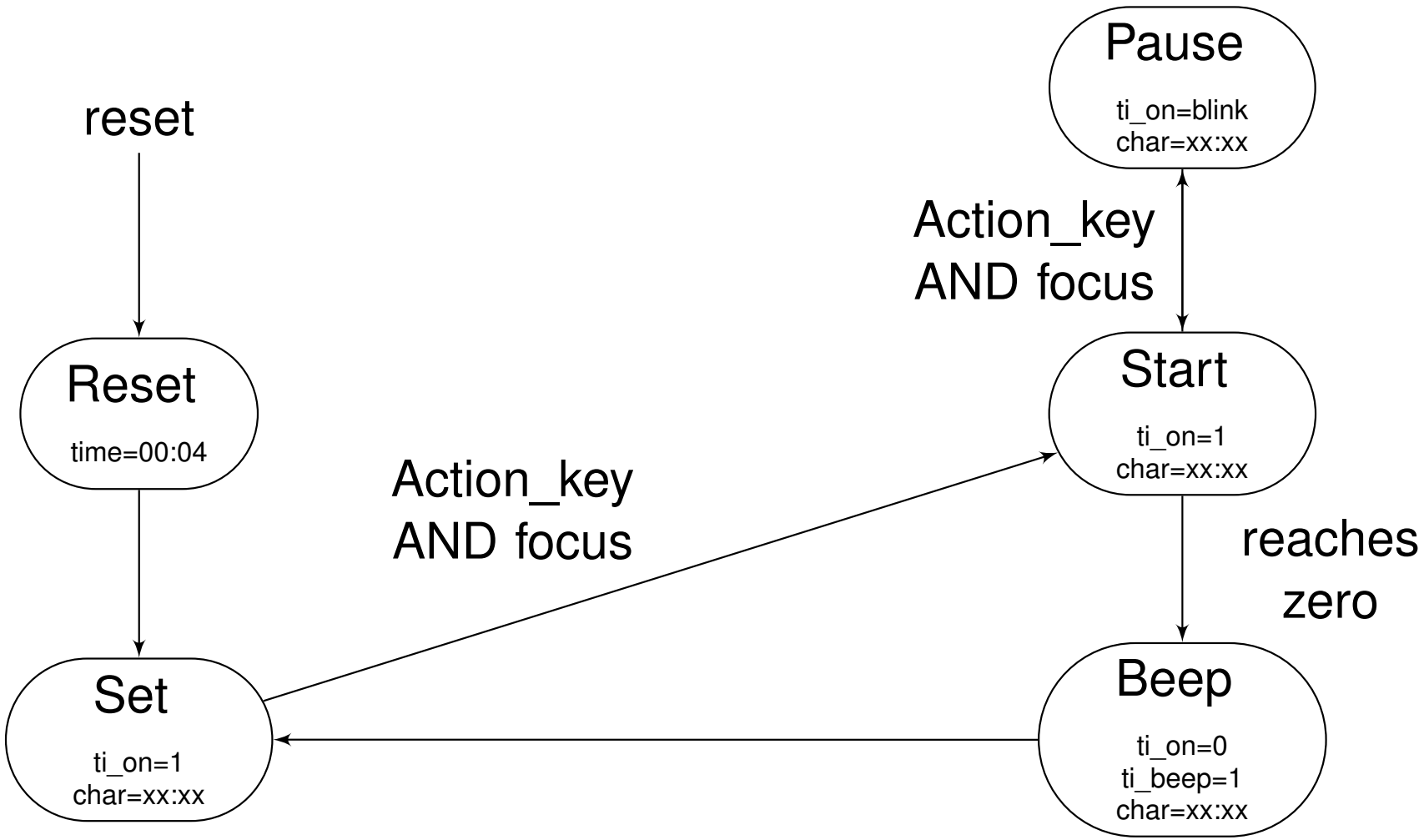
```
component countdown is
 port(
  keyboard_focus: in  std_logic_vector(2 downto 0);
  uni:            in  universal_signals;
  keys:           in  keypad_signals;

  character:      out character_array_2d(3 downto 0,
                                         19 downto 0);
  ti_on:          out std_logic;
  ti_beep:        out std_logic
 );
end component;
```

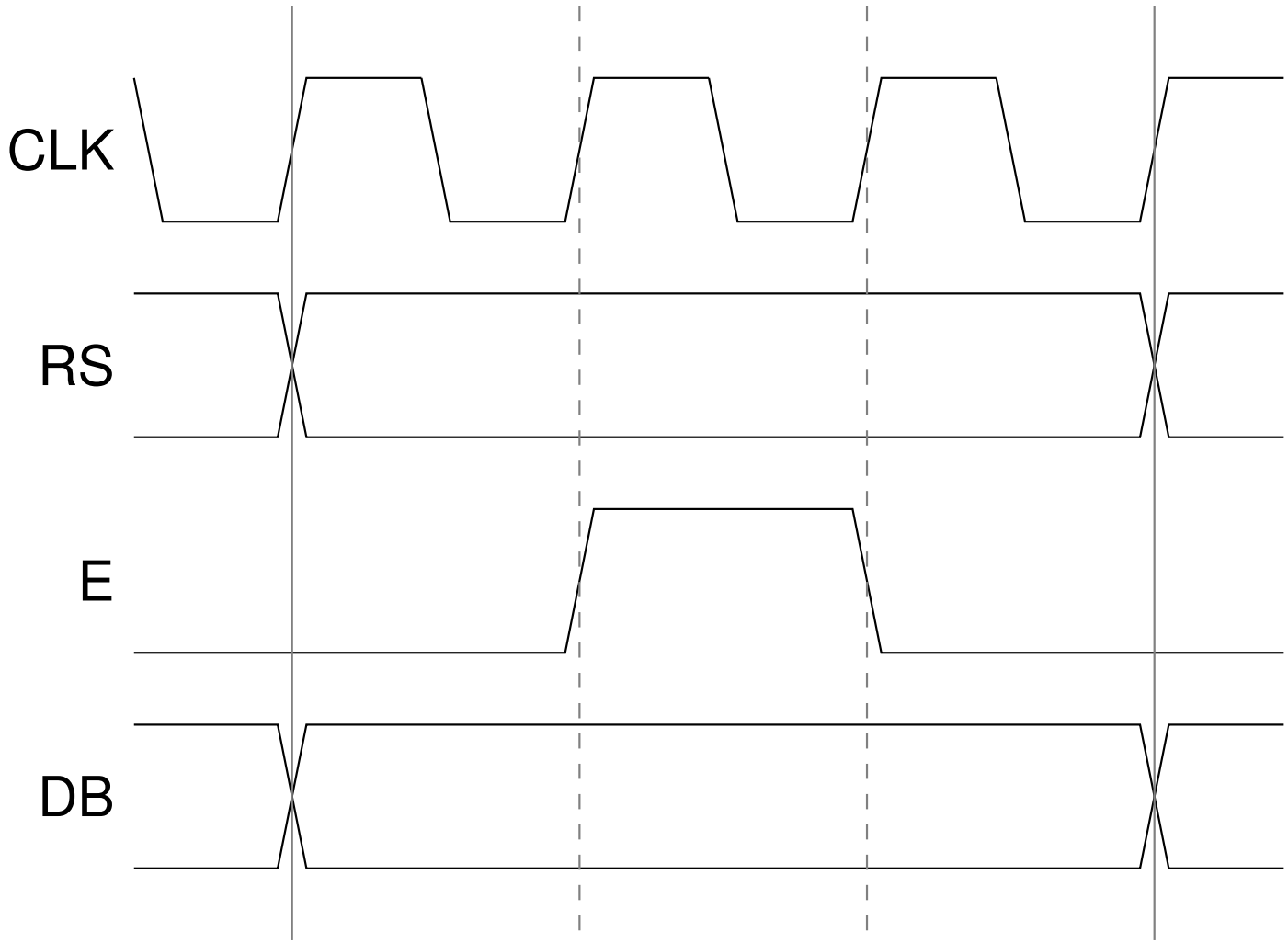
Countdown Overview



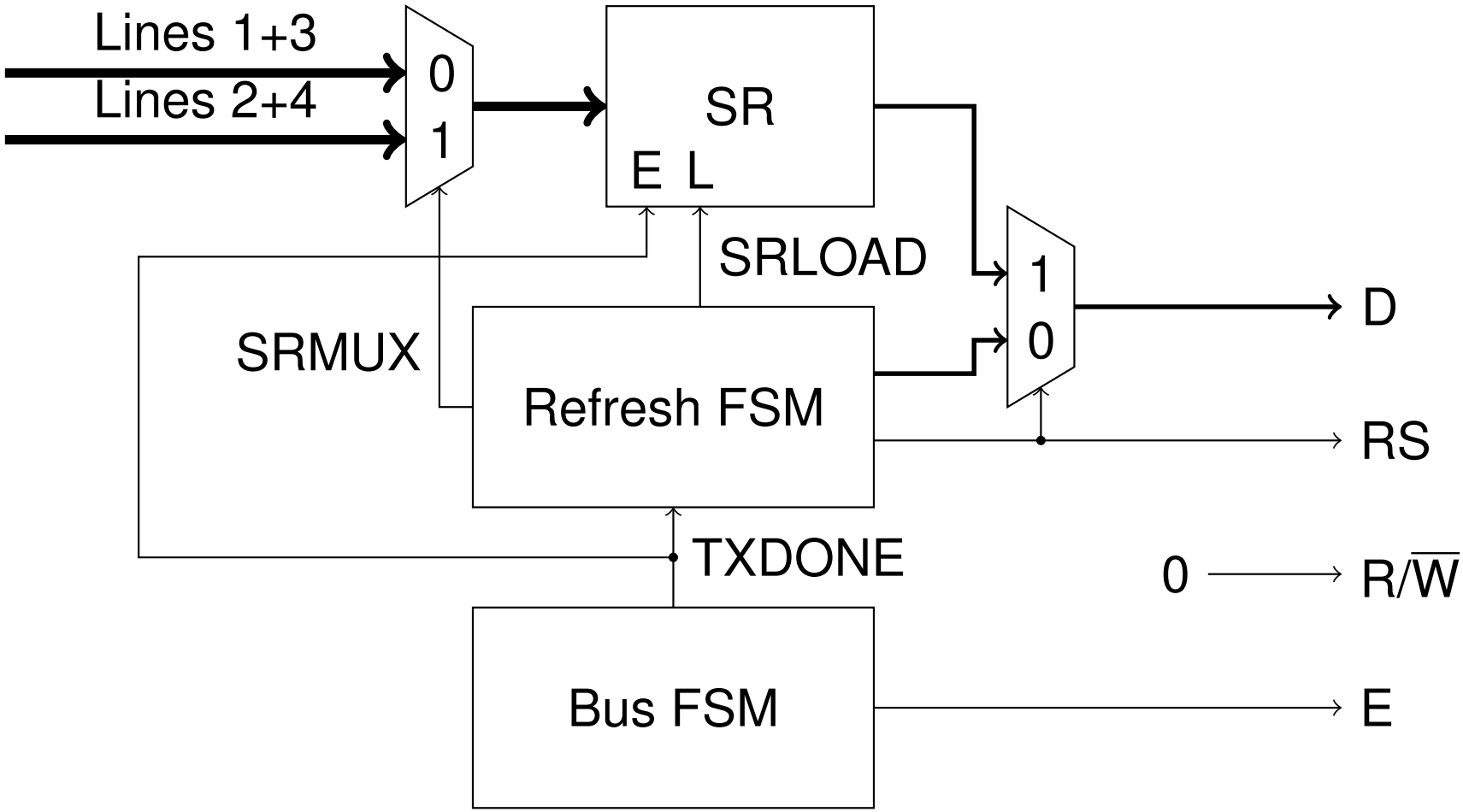
Countdown FSM



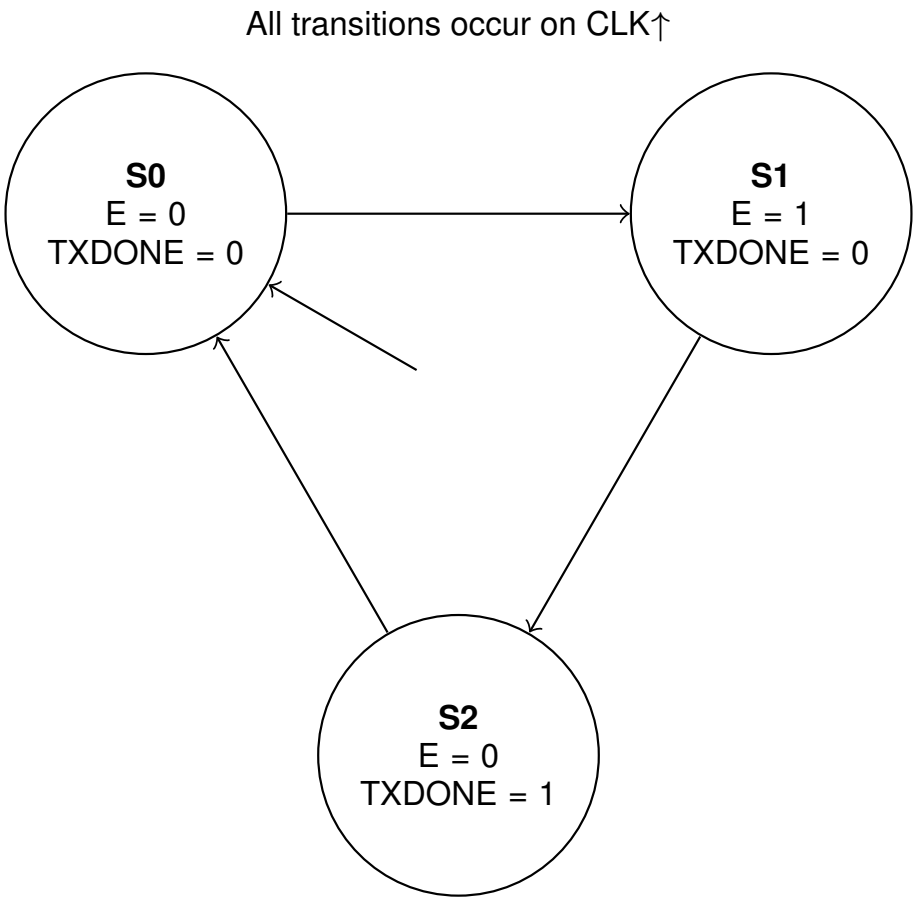
Display Timing



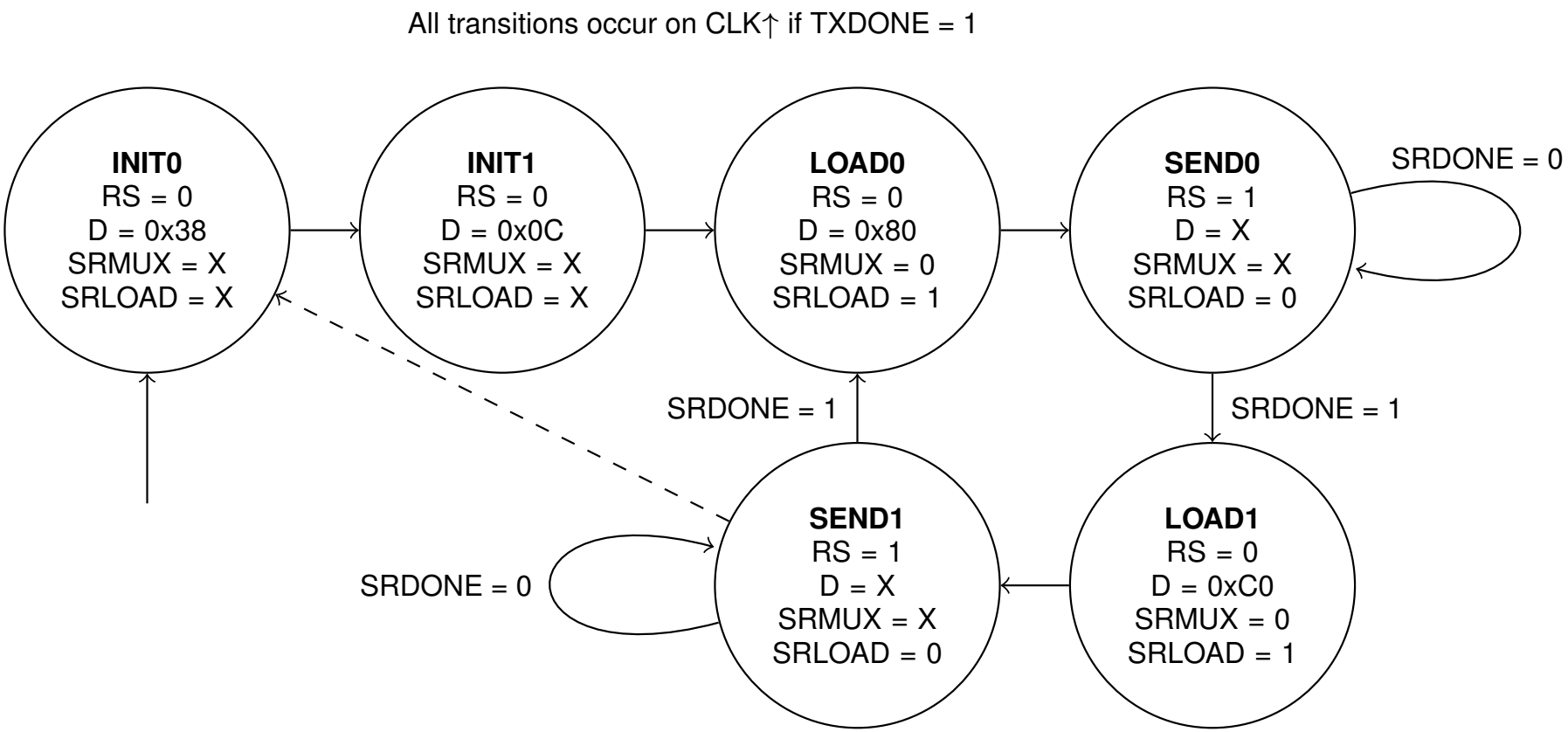
Display Driver Overview



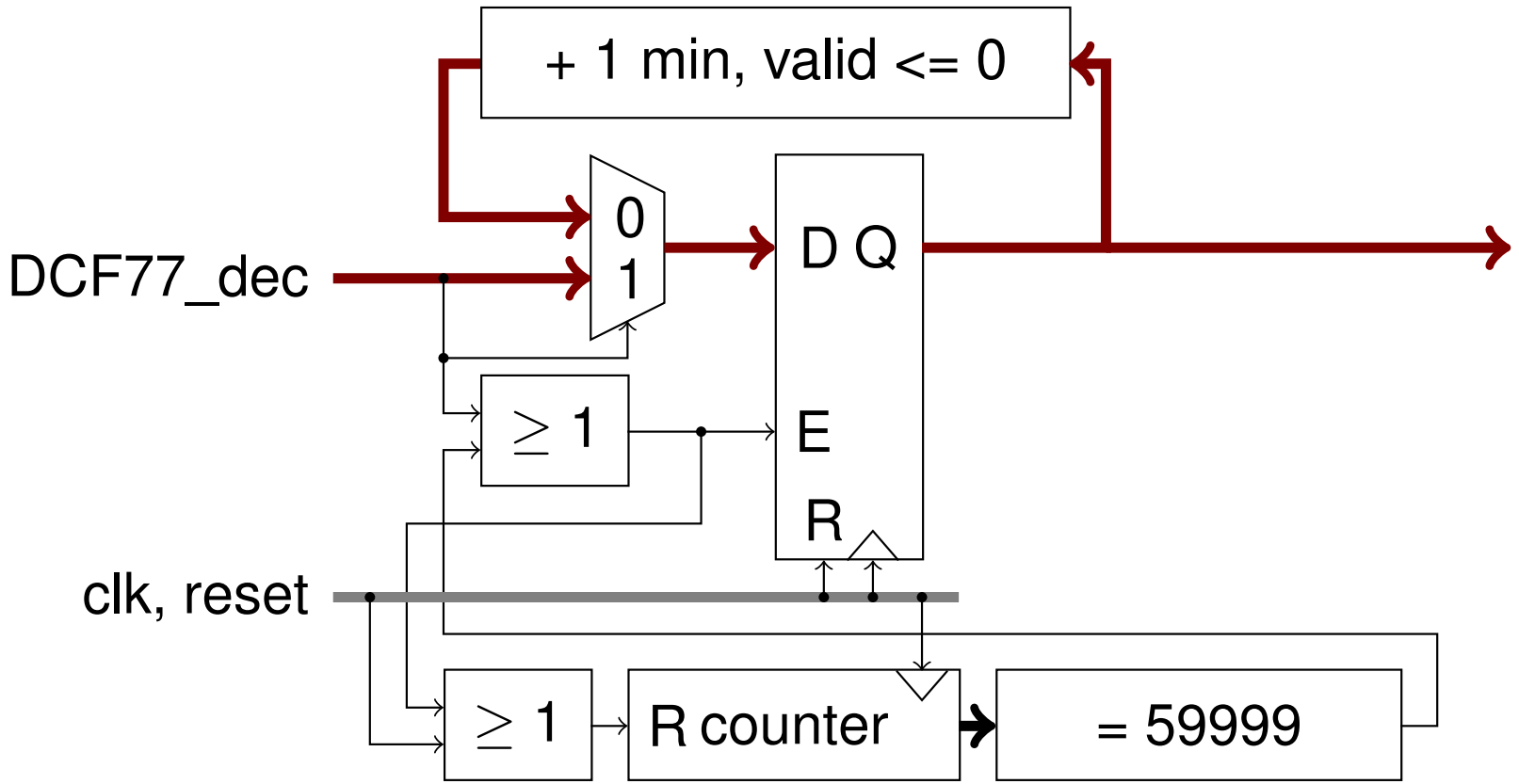
Display Driver Bus FSM



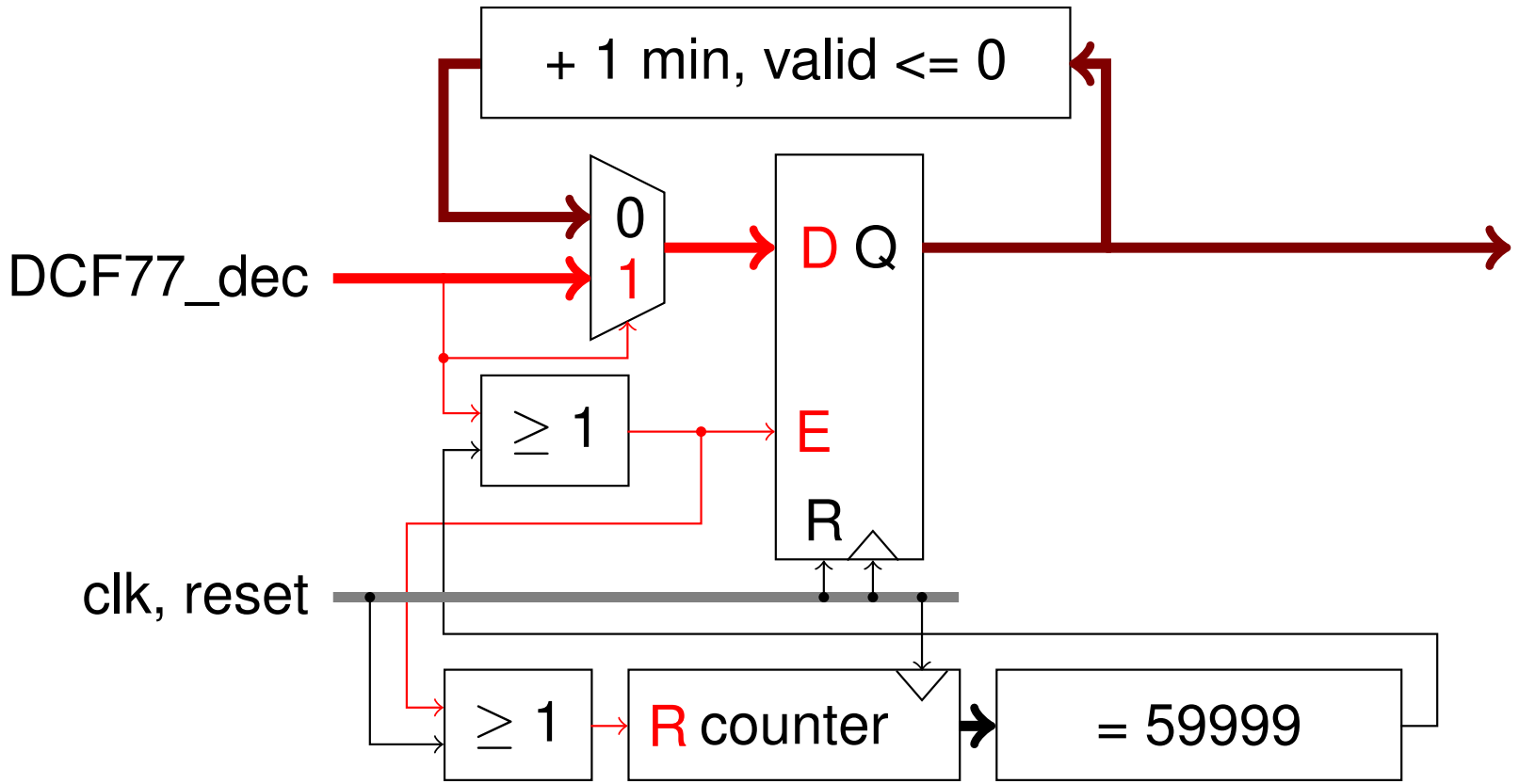
Display Driver Refresh FSM



Time Buffer Circuit



Time Buffer Circuit



Time Buffer Circuit