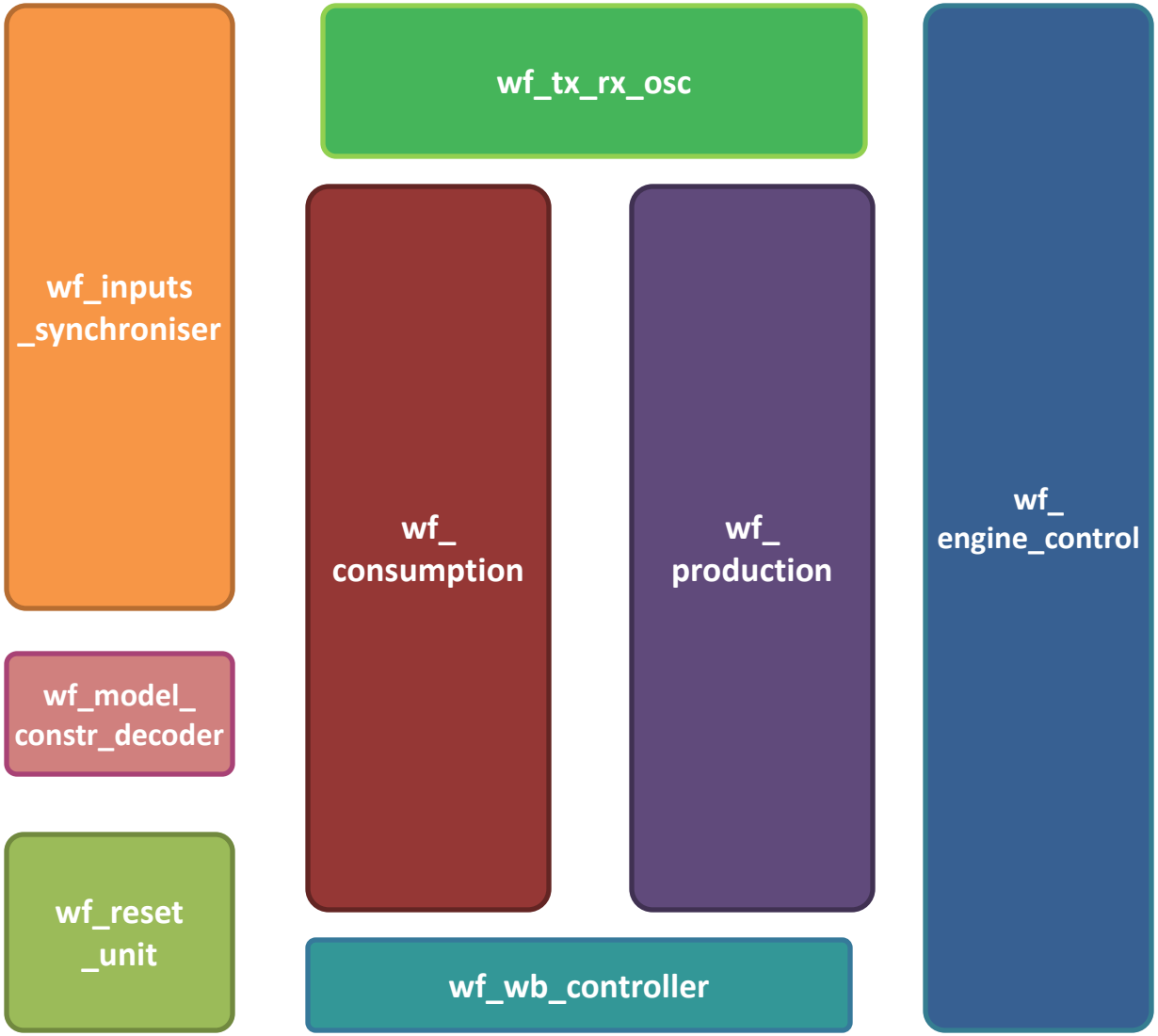
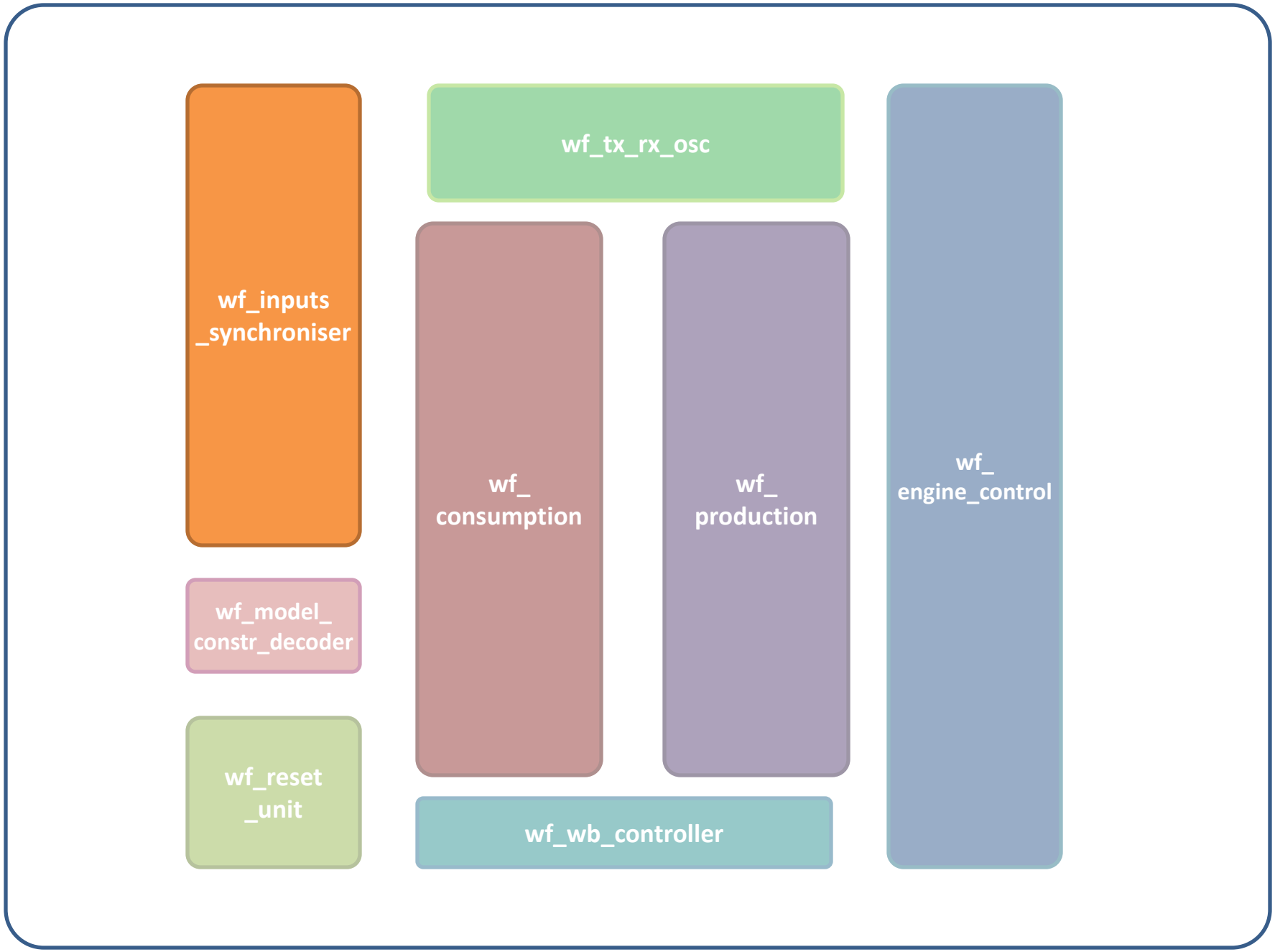


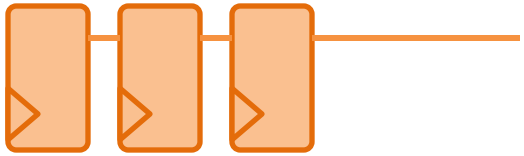
nanoFIP main blocks


wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

wf_inputs _synchroniser

Responsible for the synchronization of:

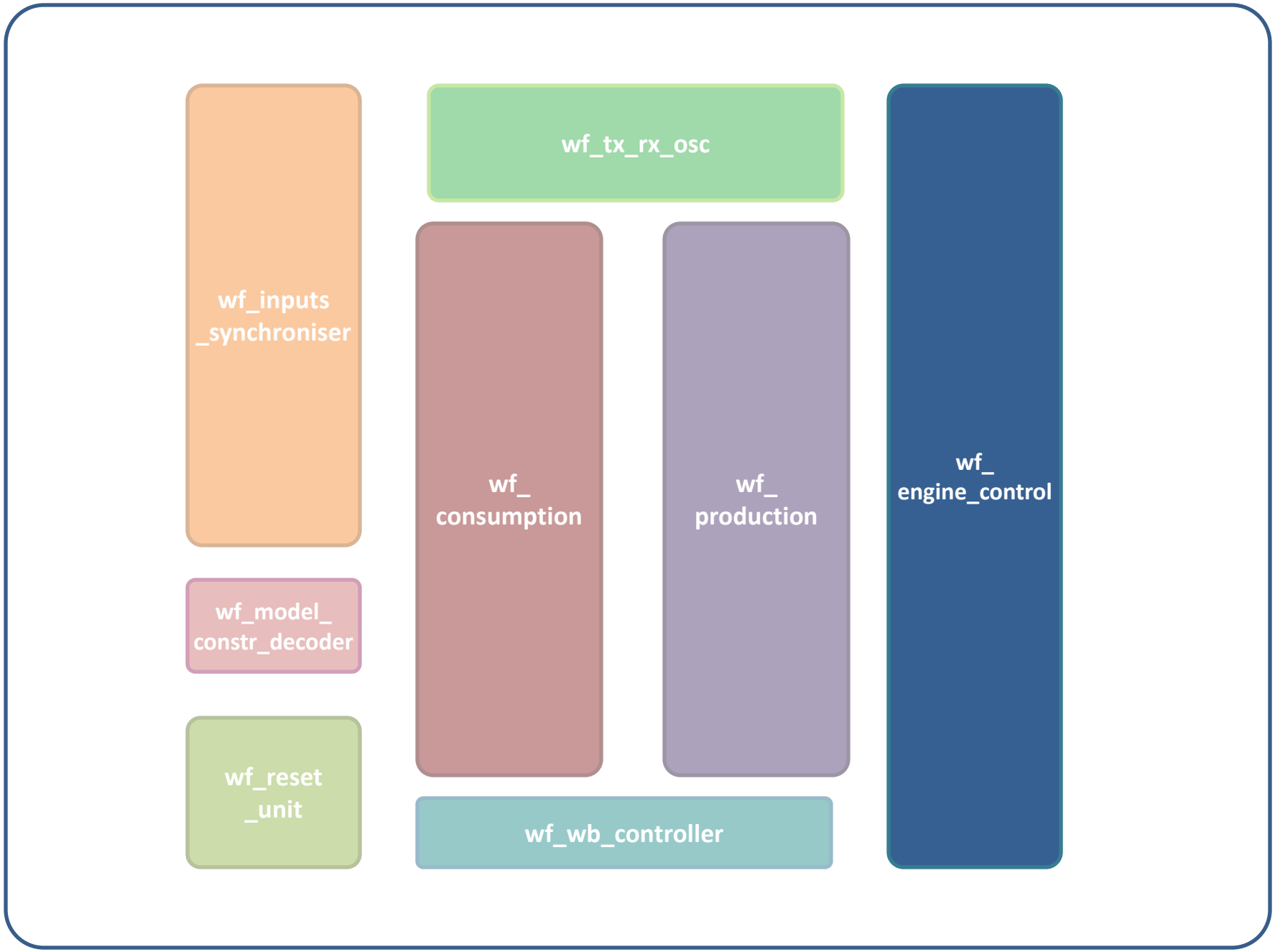
- The WISHBONE **control** signals with the wb_clk
- The FIELDRIVE signals with the uclk
- The WorldFIP settings and General signals with uclk
- The User Interface, NON WISHBONE signals with the uclk

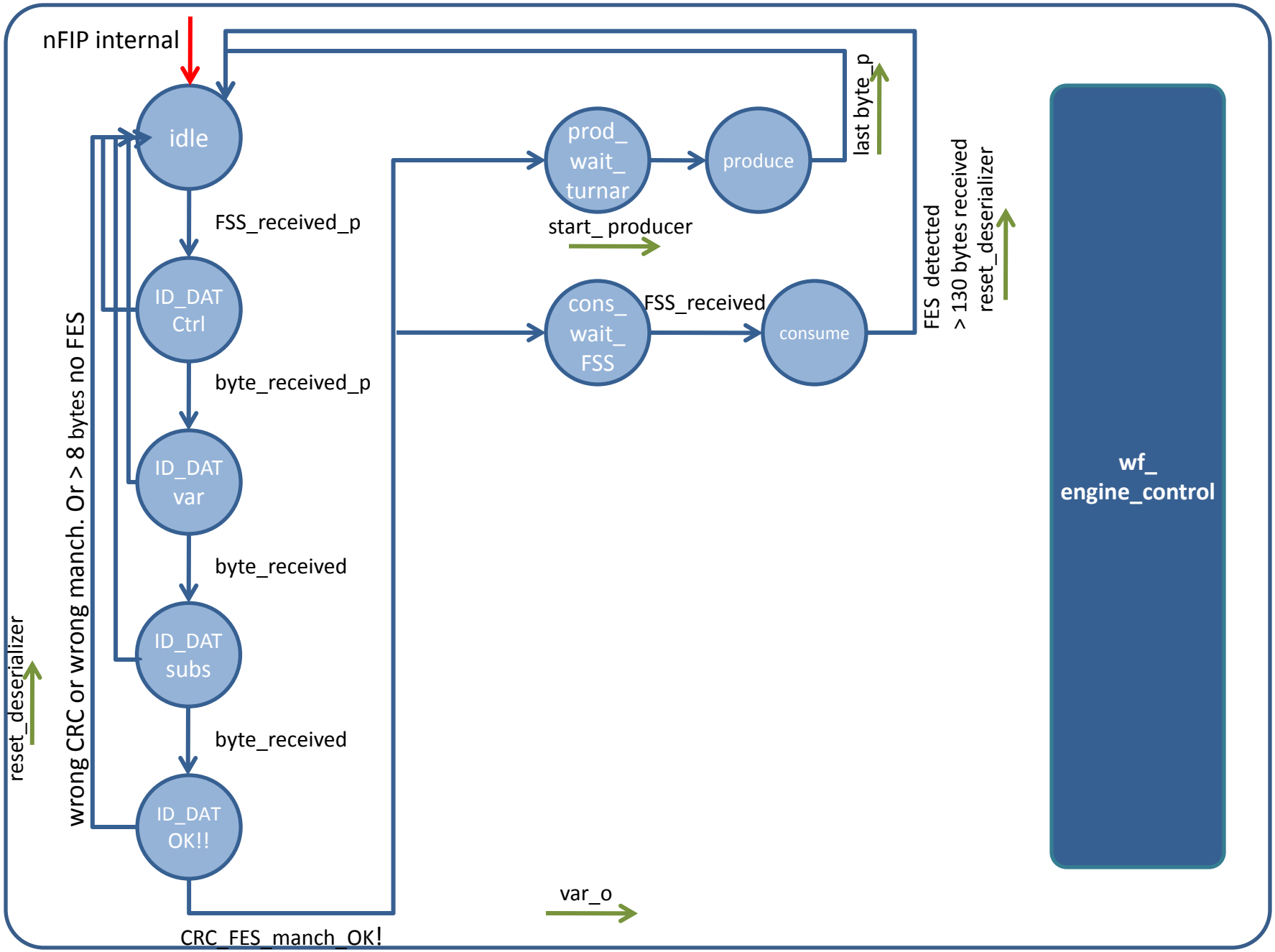


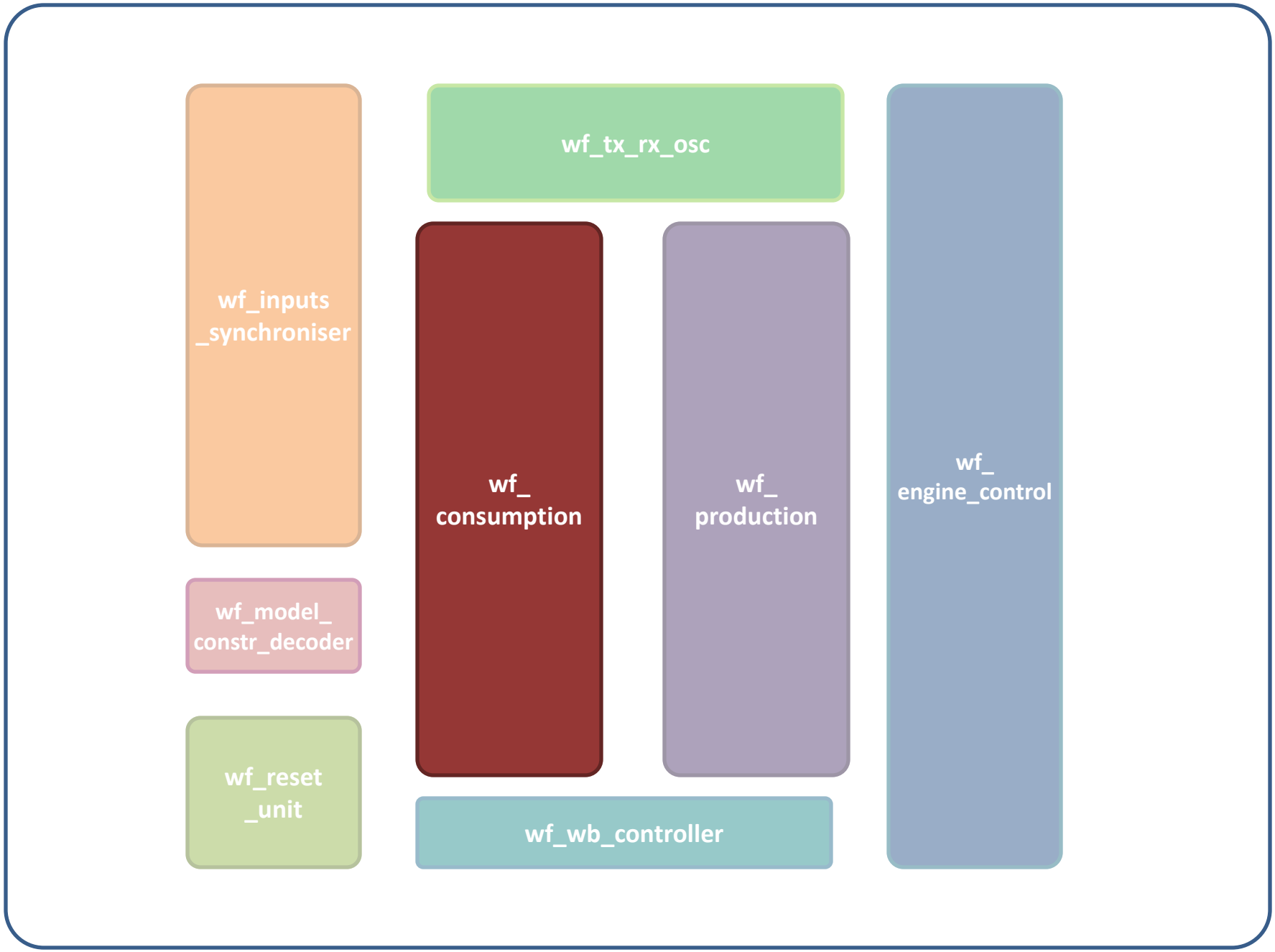
Clocks & Resets

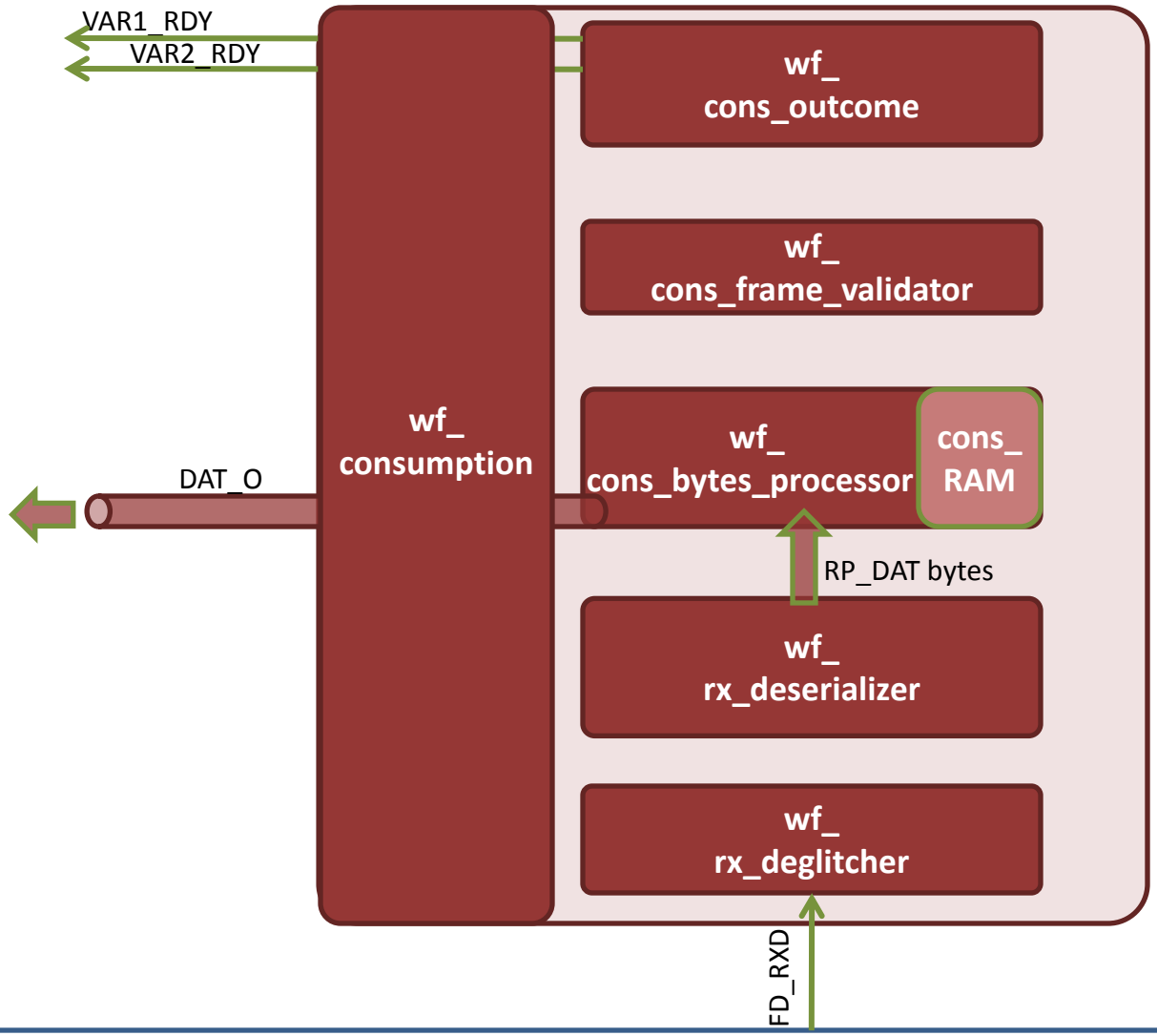
- Crossing clock domains only through the Dual Port RAM

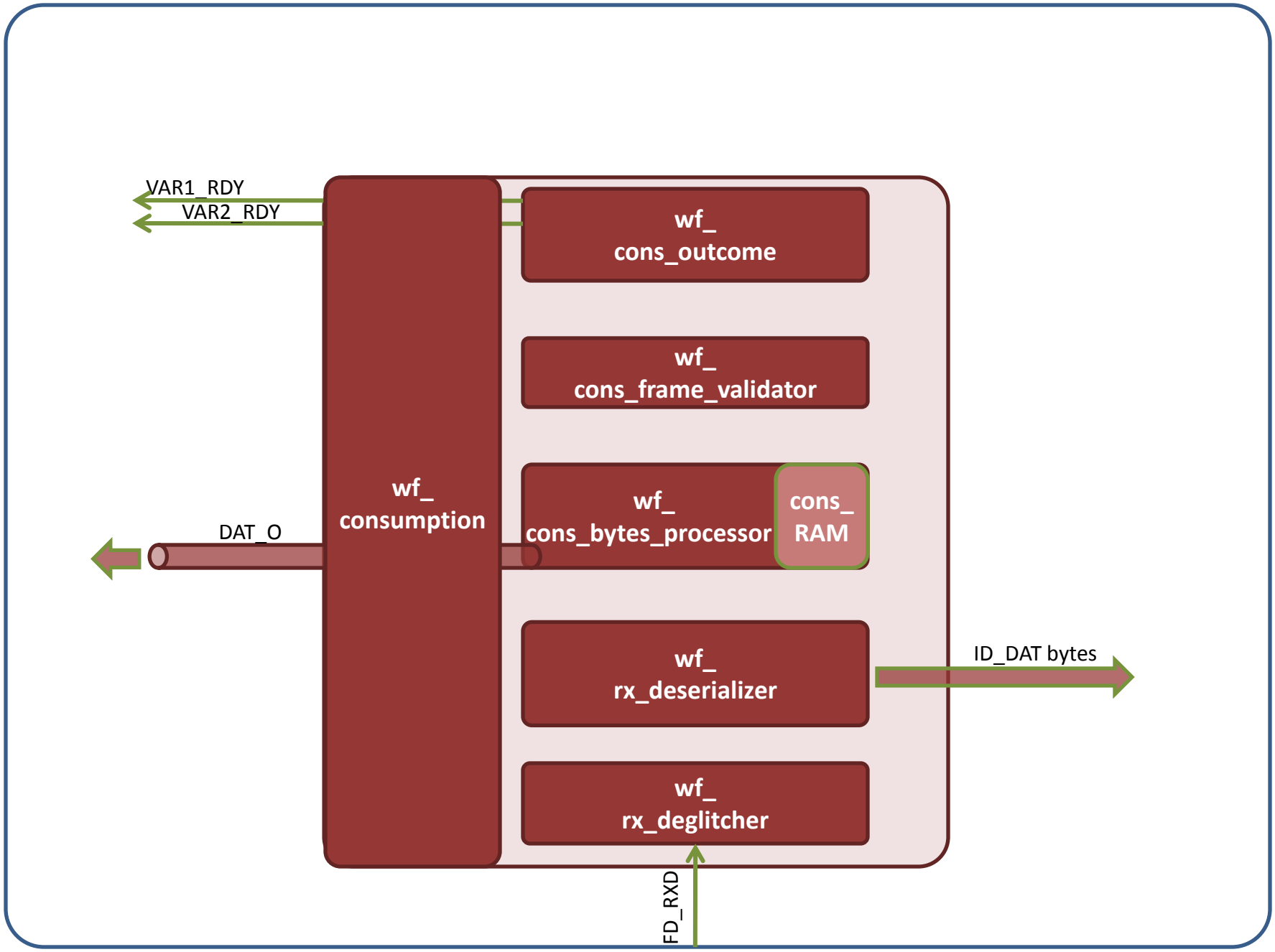
- WISHBONE reset : resets only WISHBONE logic
- nFIP internal reset: resets all the rest

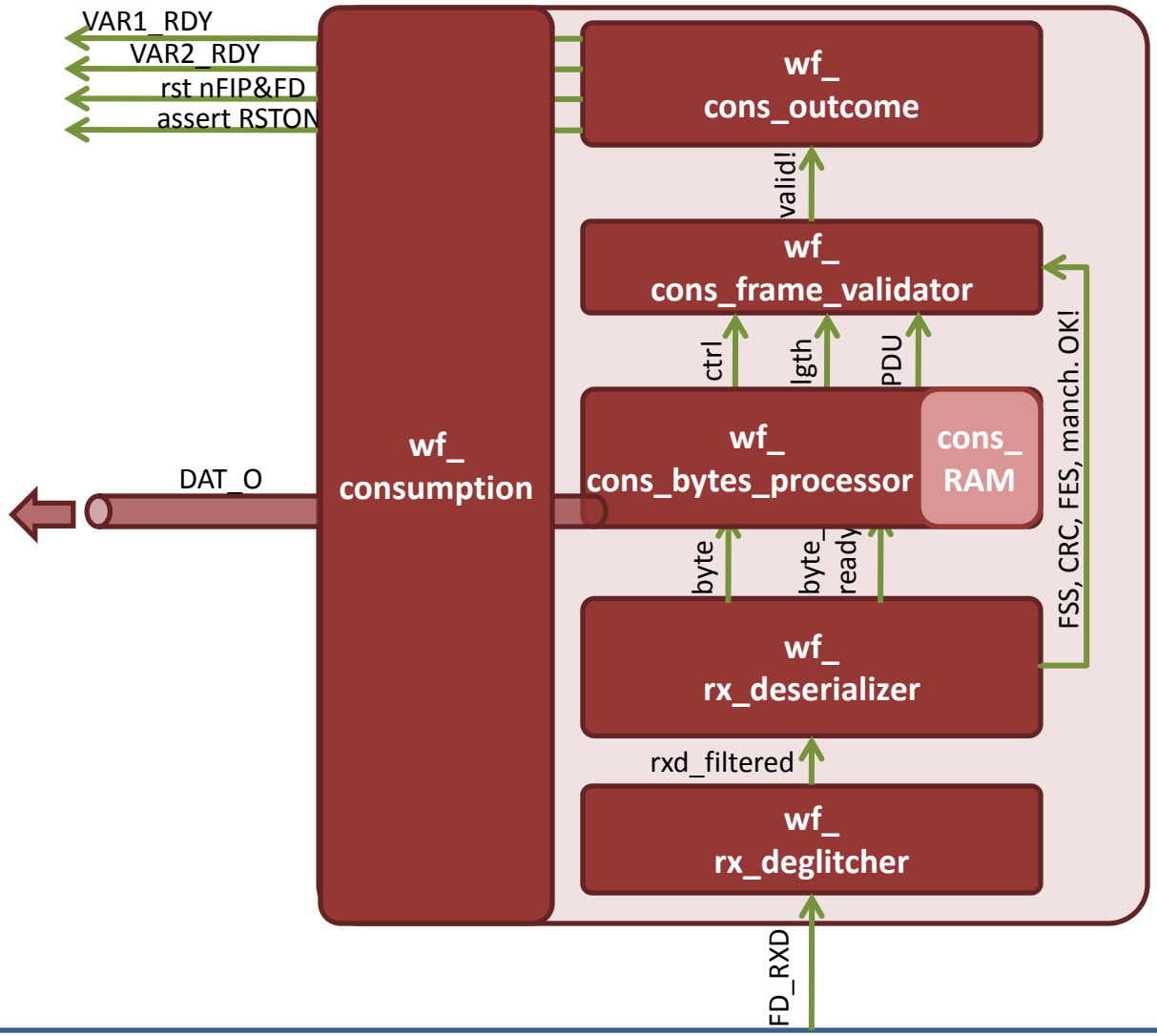

wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

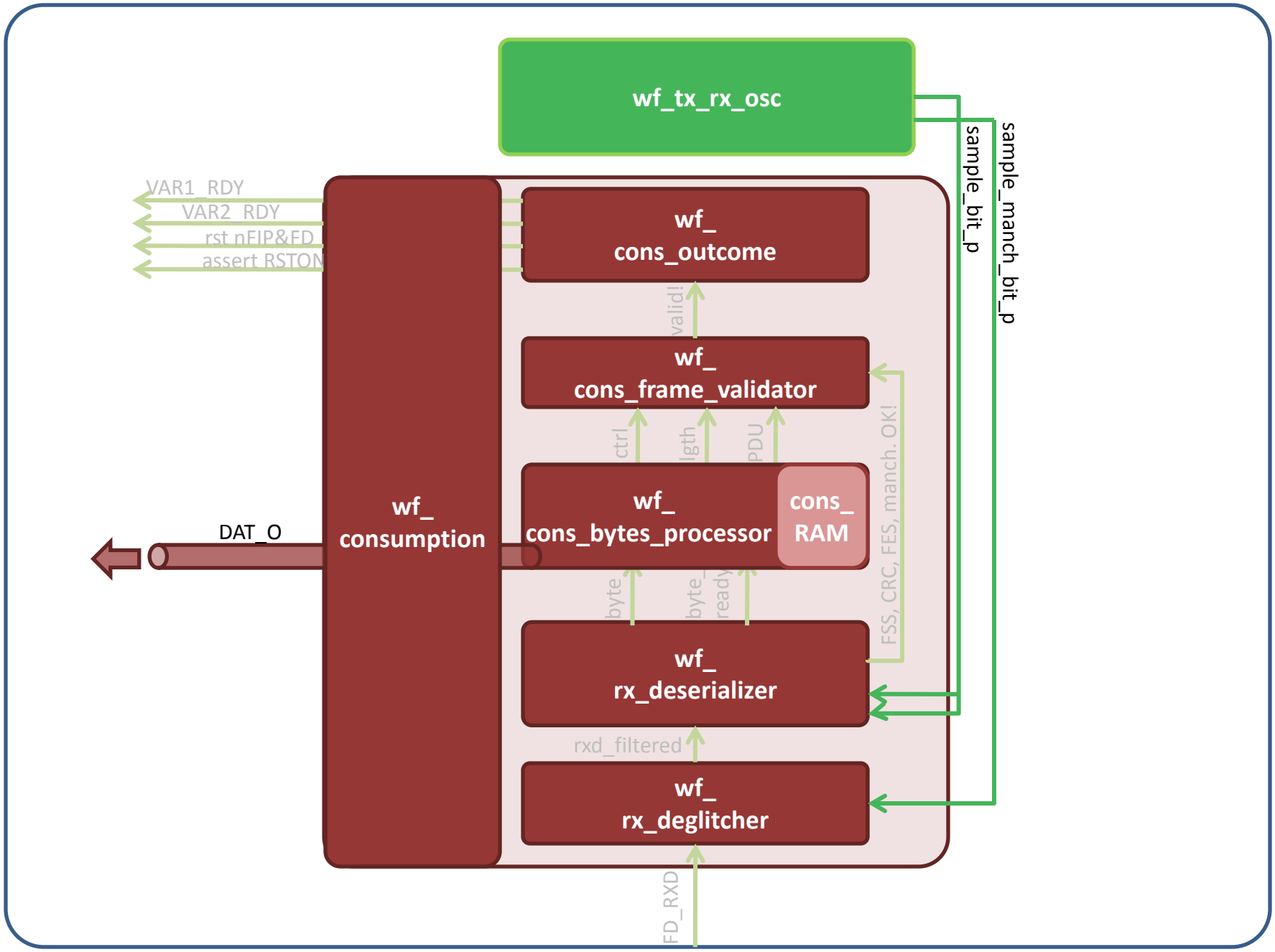

nFIP internal
idle
FSS_received_p
ID_DAT Ctrl
byte_received_p
ID_DAT var
byte_received
ID_DAT subs
byte_received
ID_DAT OK!!
CRC_FES_manch_OK!
wrong CRC or wrong manch. Or > 8 bytes no FES
reset_deserializer
prod_ wait_ turnar
produce
last byte_p
start_ producer
cons_ wait_ FSS
FSS_received
consume
FES detected
> 130 bytes received
reset_deserializer
wf_ engine_control
var_o

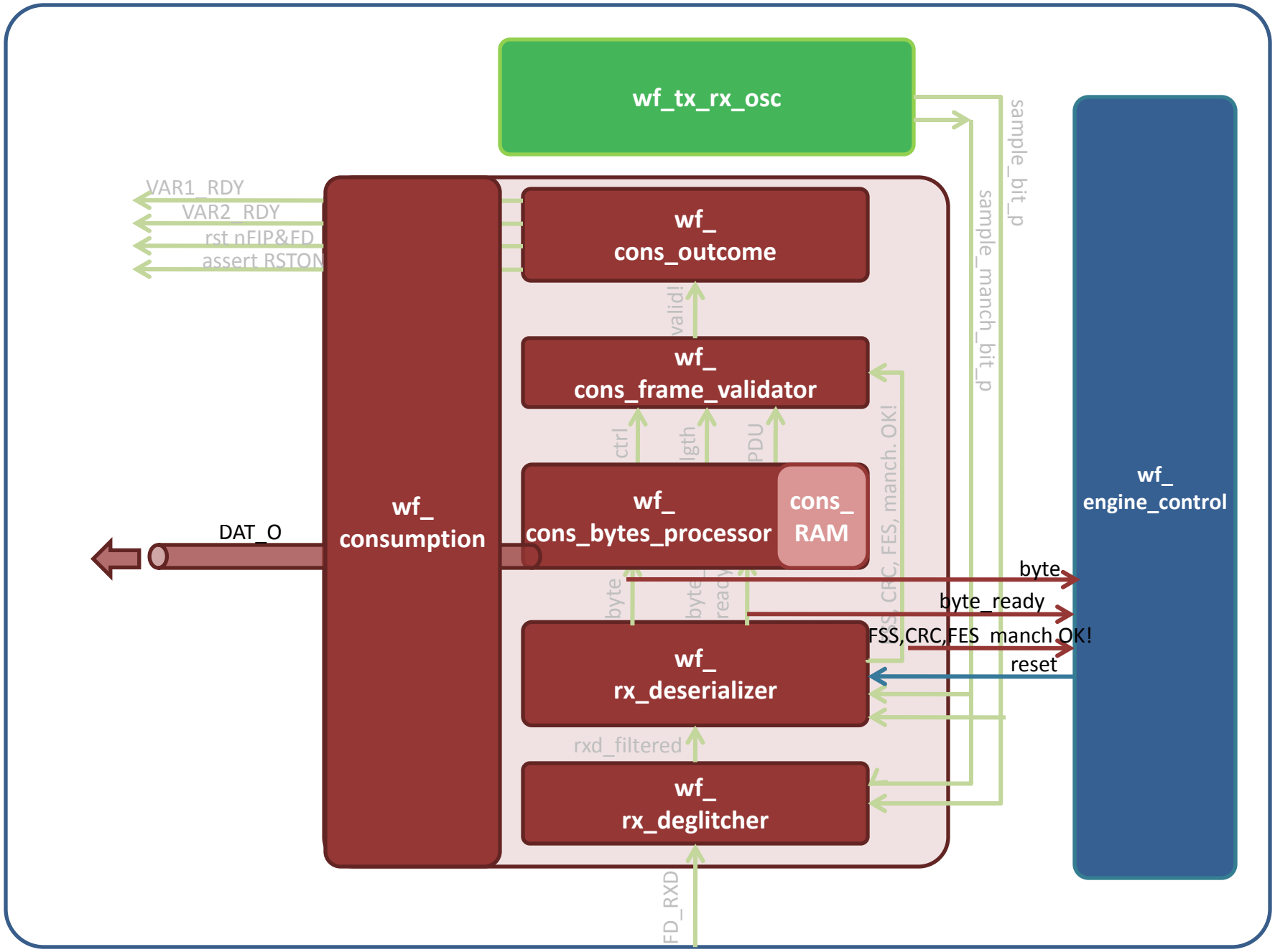

wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

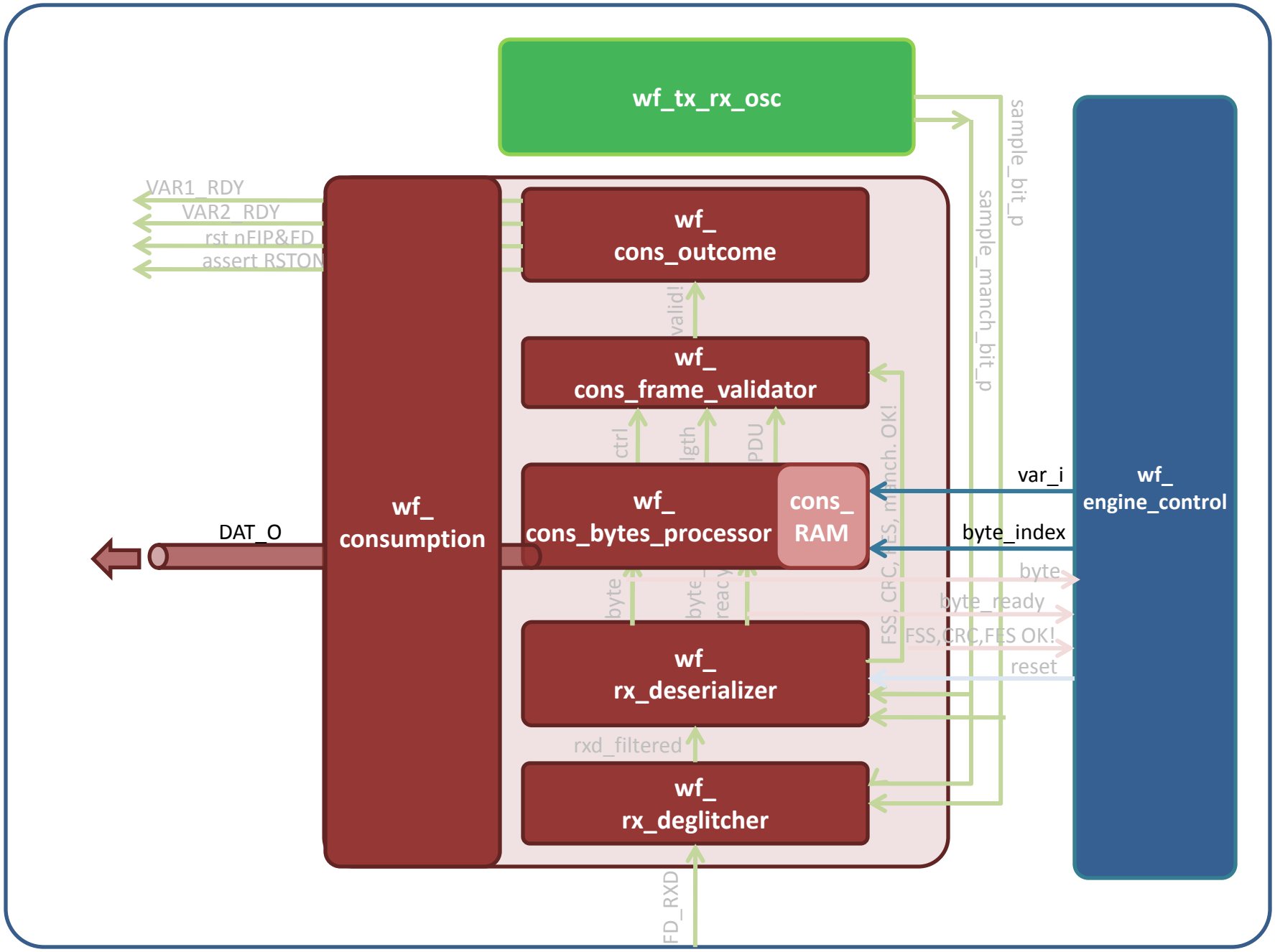

VAR1_RDY
VAR2_RDY
wf_
cons_outcome
wf_
cons_frame_validator
wf_
consumption
wf_
cons_bytes_processor
cons_
RAM
DAT_O
RP_DAT bytes
wf_
rx_deserializer
wf_
rx_deglitcher
FD_RXD

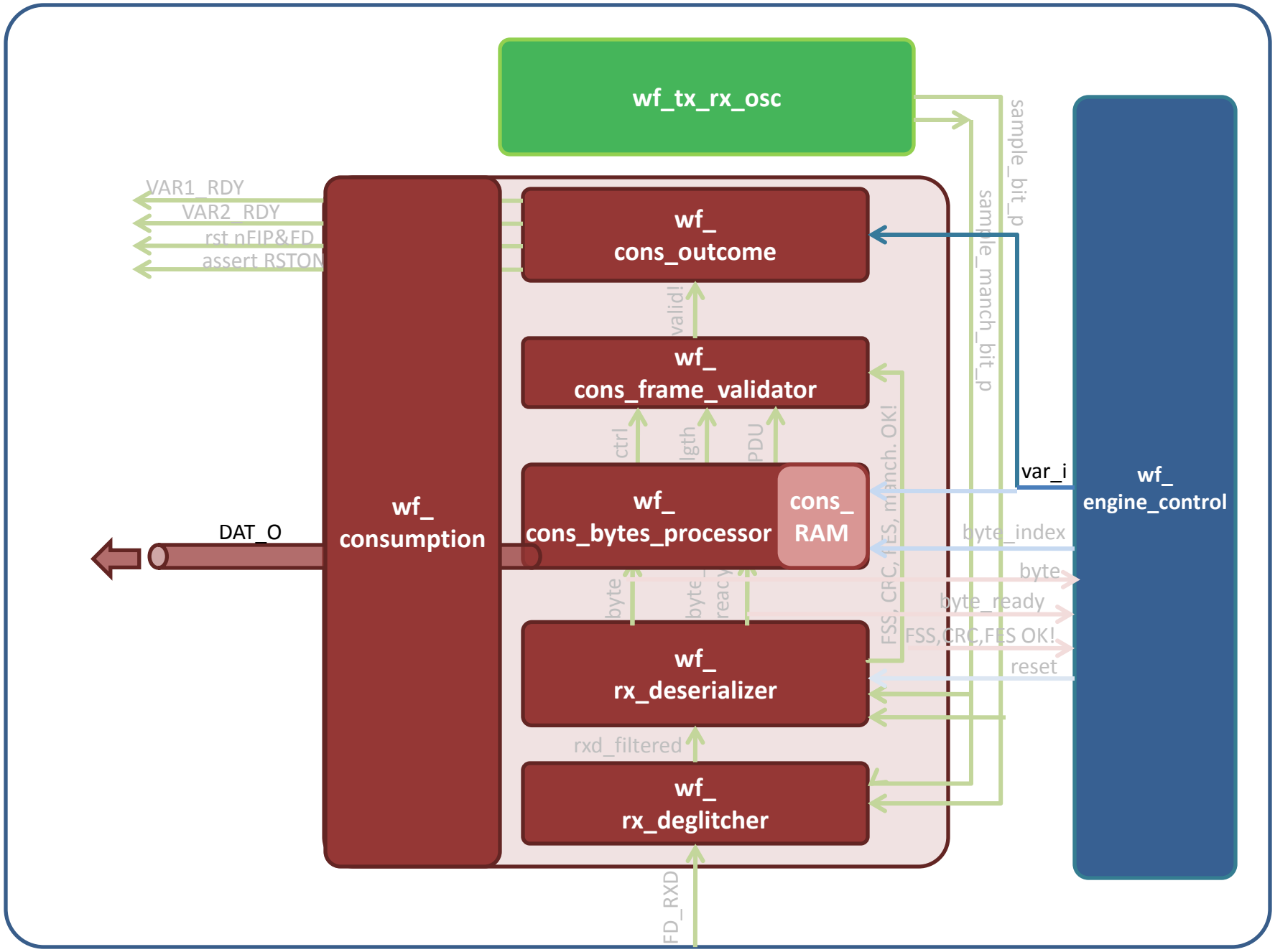

VAR1_RDY
VAR2_RDY
wf_
cons_outcome
wf_
cons_frame_validator
wf_
consumption
wf_
cons_bytes_processor
cons_
RAM
DAT_O
wf_
rx_deserializer
ID_DAT bytes
wf_
rx_deglitcher
FD_RXD

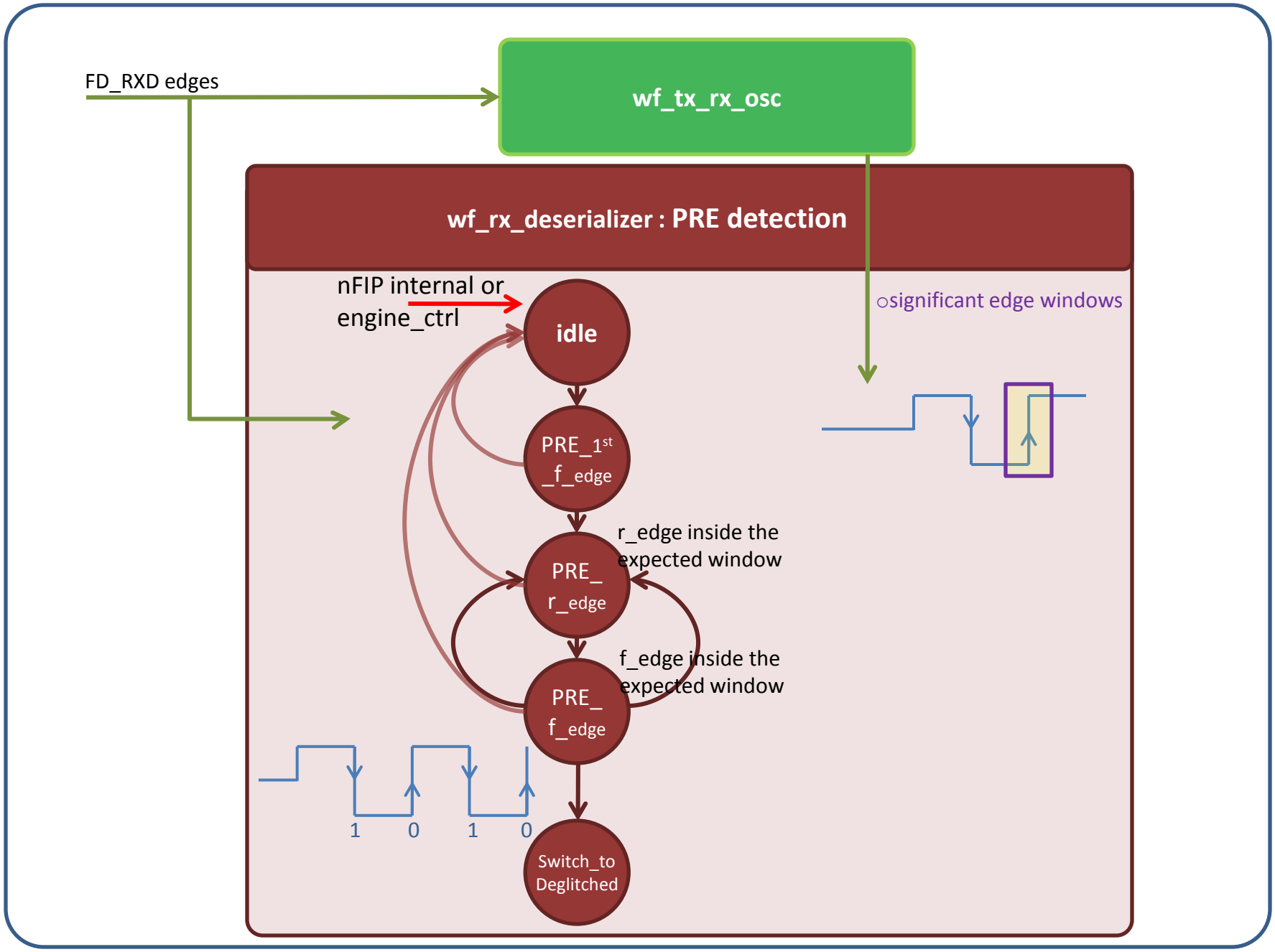

VAR1_RDY
VAR2_RDY
rst nFIP&FD
assert RSTON
DAT_O
wf_
consumption
wf_
cons_outcome
valid!
wf_
cons_frame_validator
ctrl
lgth
PDU
wf_
cons_bytes_processor
cons_
RAM
byte
byte_
ready
FSS, CRC, FES, manch. OK!
wf_
rx_deserializer
rxd_filtered
wf_
rx_deglitcher
FD_RXD

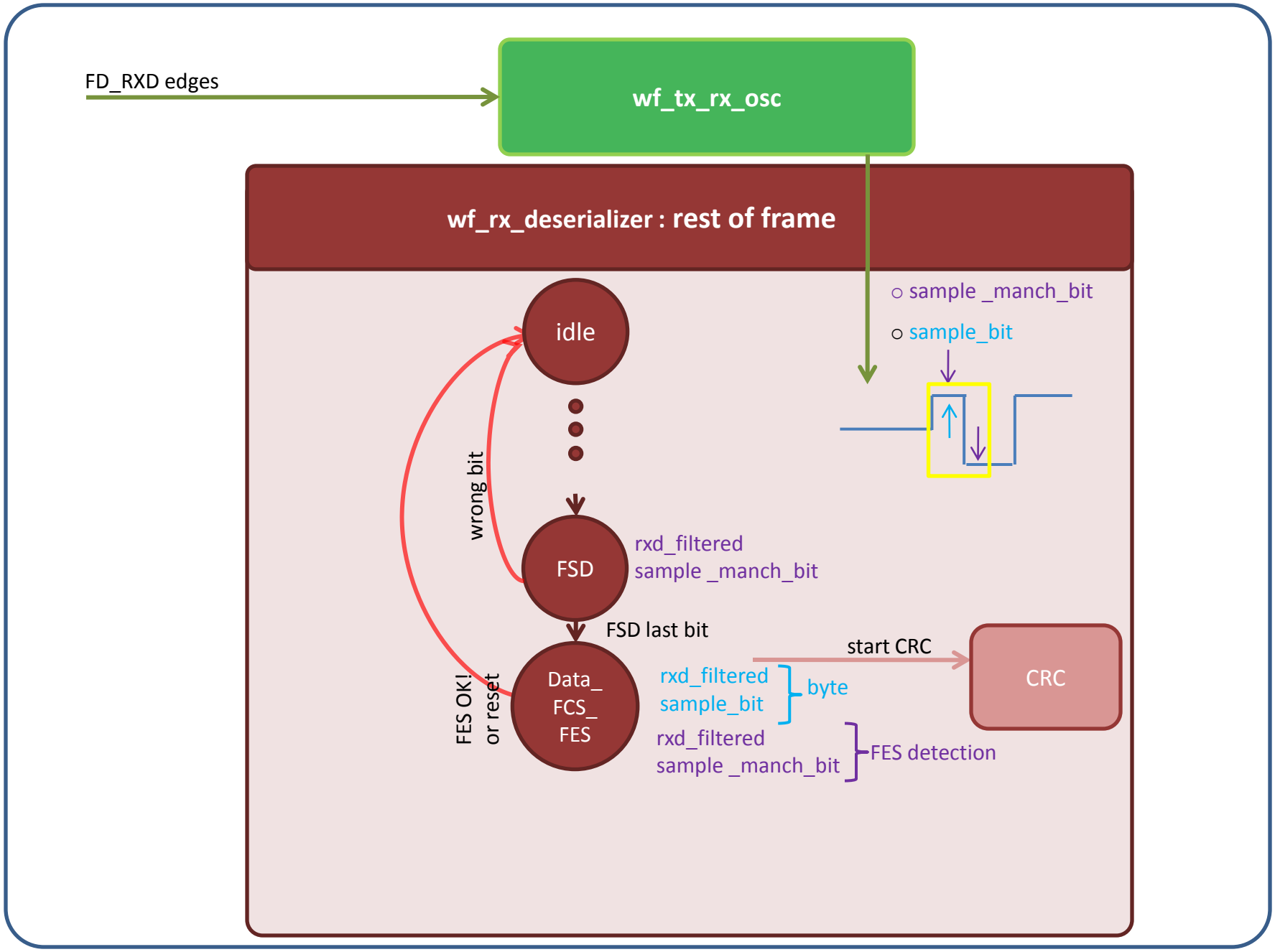

wf_tx_rx_osc
sample_bit_p
sample_manch_bit_p
VAR1_RDY
VAR2_RDY
rst nFIP&FD
assert RSTON
wf_
cons_outcome
valid!
wf_
cons_frame_validator
ctrl
lgth
PDU
FSS, CRC, FES, manch. OK!
wf_
cons_bytes_processor
cons_
RAM
DAT_O
wf_
consumption
byte
byte_
ready
wf_
rx_deserializer
rxd_filtered
wf_
rx_deglitcher
FD_RXD

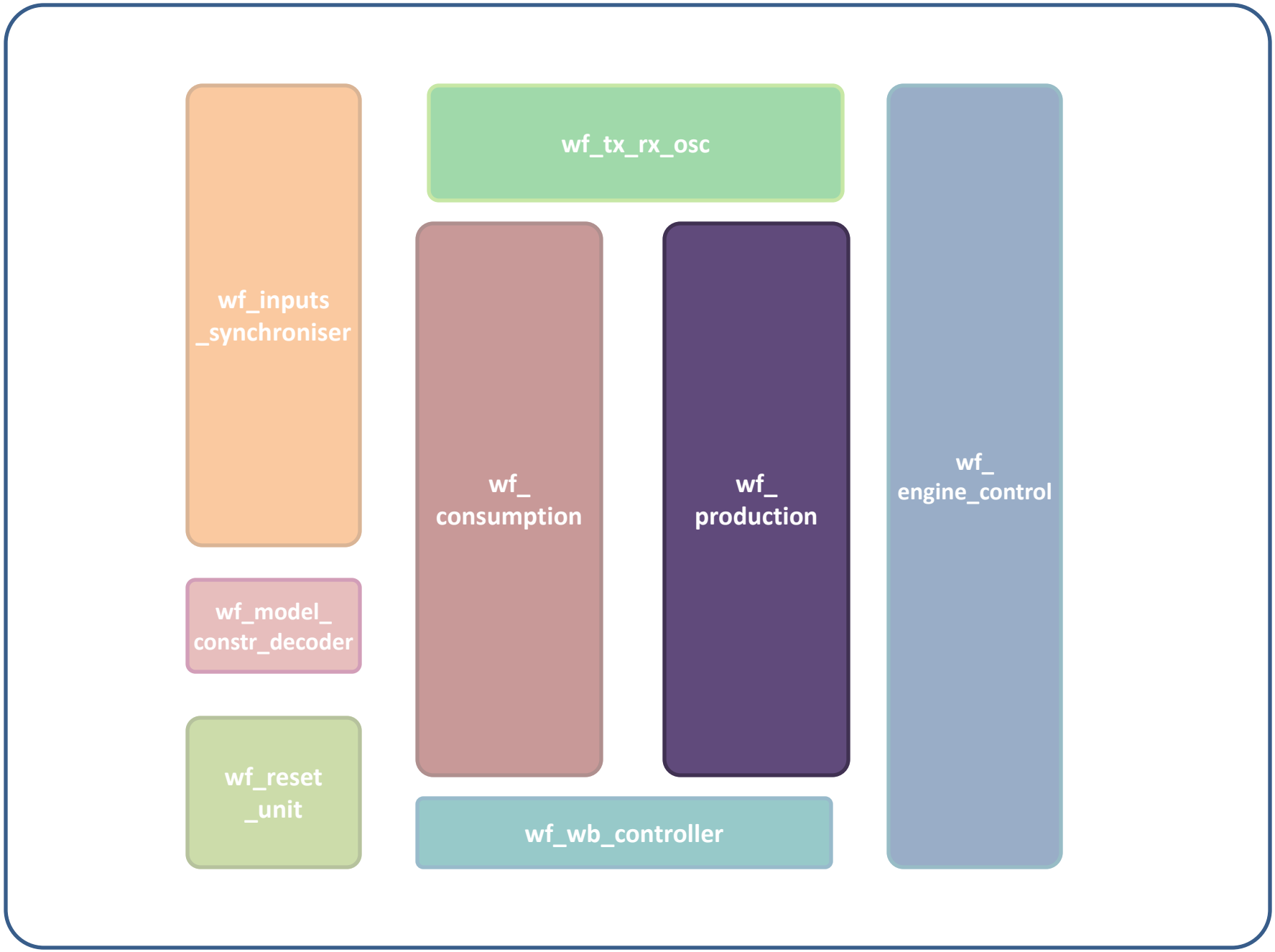

wf_tx_rx_osc
sample_bit_p
sample_manch_bit_p
VAR1_RDY
VAR2_RDY
rst nFIP&FD
assert RSTON
wf_
cons_outcome
valid!
wf_
cons_frame_validator
ctrl
lgth
PDU
wf_
consumption
wf_
cons_bytes_processor
cons_
RAM
DAT_O
byte
byte_
ready
FSS, CRC, FES, manch. OK!
wf_
rx_deserializer
rxd_filtered
wf_
rx_deglitcher
FD_RXD
wf_
engine_control
byte
byte_ready
FSS,CRC,FES manch OK!
reset

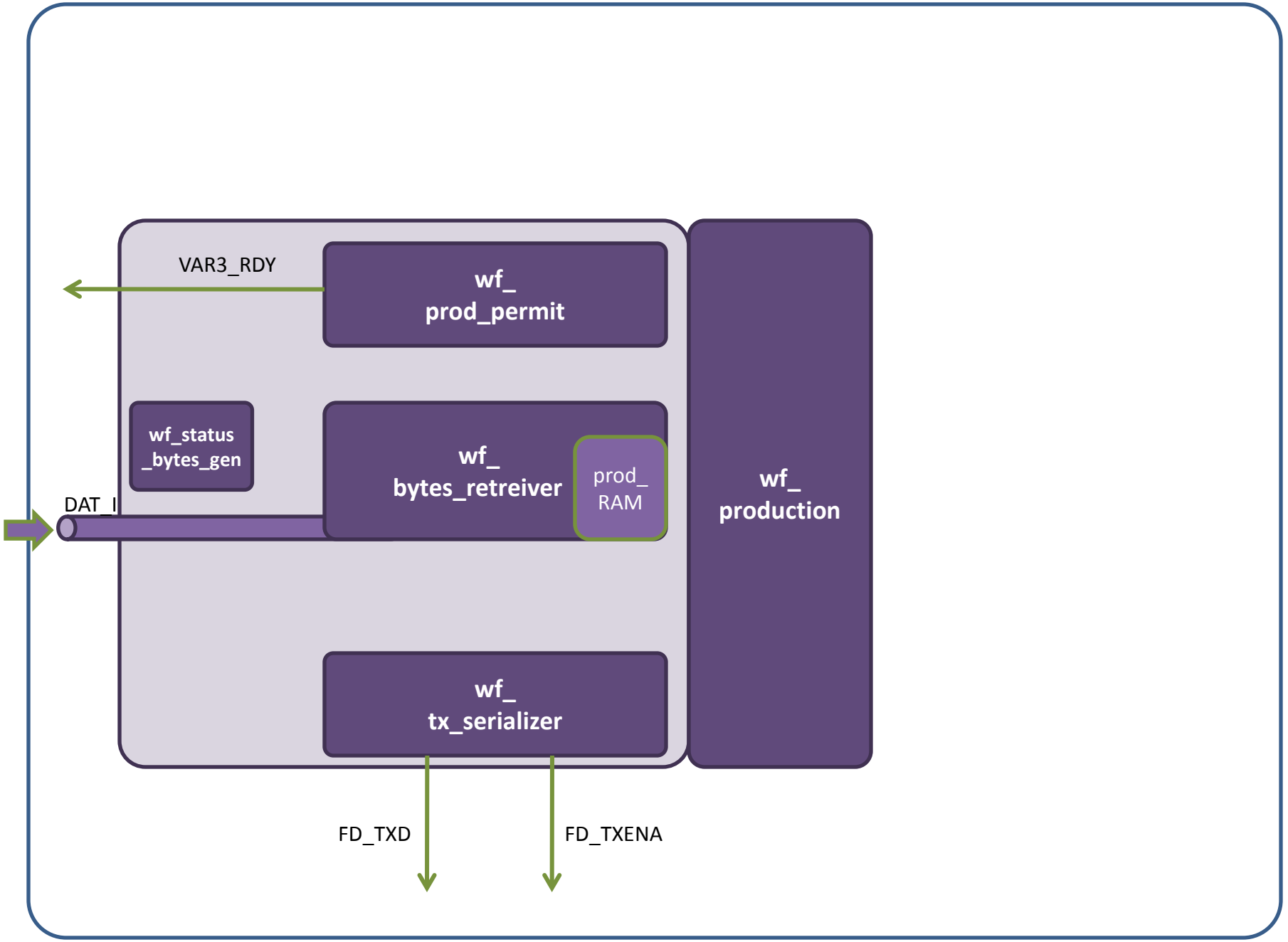

wf_tx_rx_osc
sample_bit_p
sample_manch_bit_p
VAR1_RDY
VAR2_RDY
rst nFIP&FD
assert RSTON
wf_
cons_outcome
valid!
wf_
cons_frame_validator
ctrl
lgth
PDU
wf_
cons_bytes_processor
cons_
RAM
FSS, CRC, FES, manch. OK!
wf_
consumption
DAT_O
var_i
byte_index
byte
byte_ready
FSS,CRC,FES OK!
reset
wf_
engine_control
byte
byte_
reacy
wf_
rx_deserializer
rxd_filtered
wf_
rx_deglitcher
FD_RXD

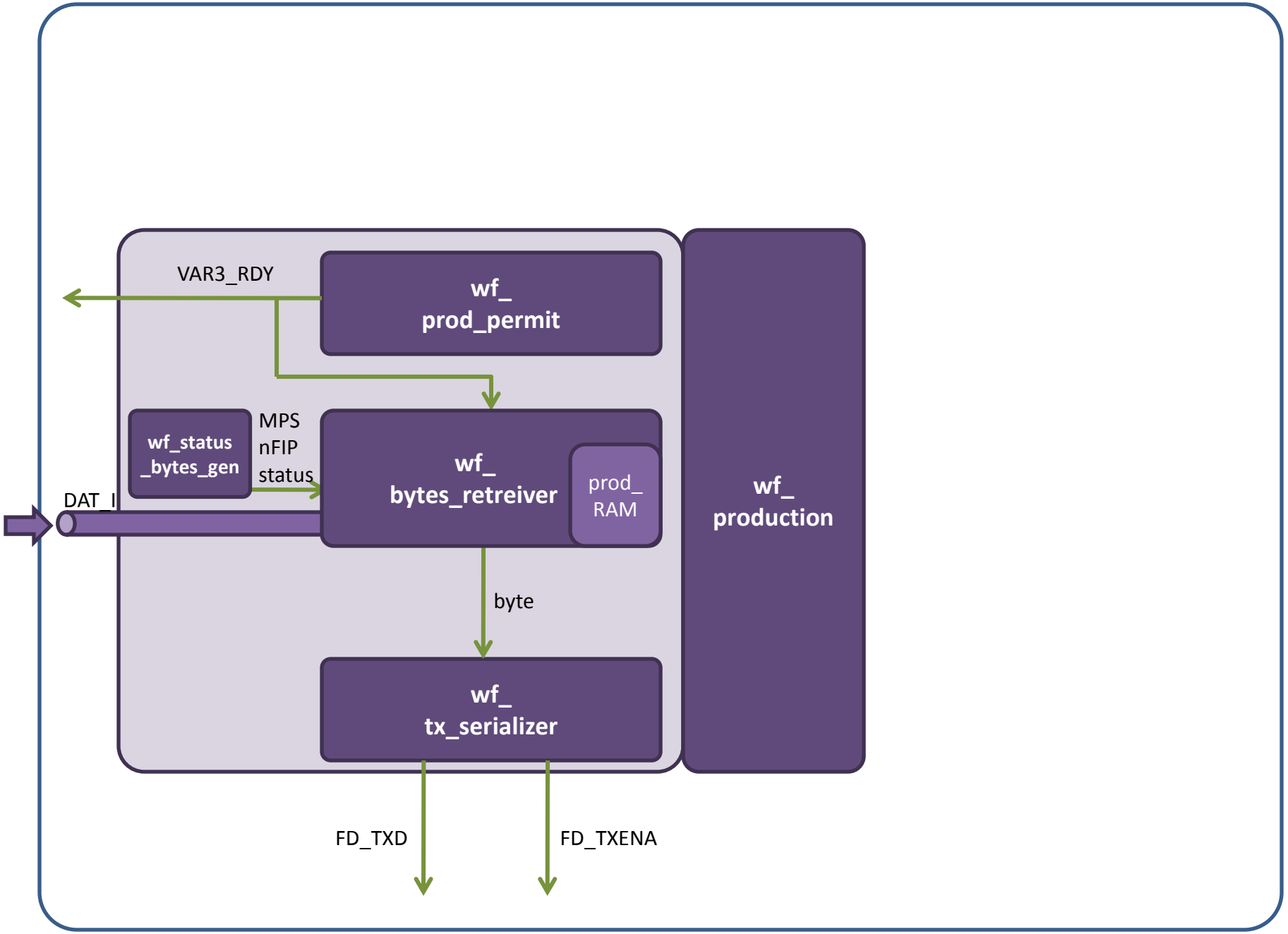

wf_tx_rx_osc
sample_bit_p
sample_manch_bit_p
wf_
engine_control
VAR1_RDY
VAR2_RDY
rst nFIP&FD
assert RSTON
wf_
cons_outcome
valid!
wf_
cons_frame_validator
ctrl
lgth
PDU
wf_
consumption
wf_
cons_bytes_processor
cons_
RAM
FSS, CRC, FES, manch. OK!
var_i
byte_index
byte
byte_ready
FSS,CRC,FES OK!
reset
DAT_O
byte
byte_
reacy
wf_
rx_deserializer
rxd_filtered
wf_
rx_deglitcher
FD_RXD

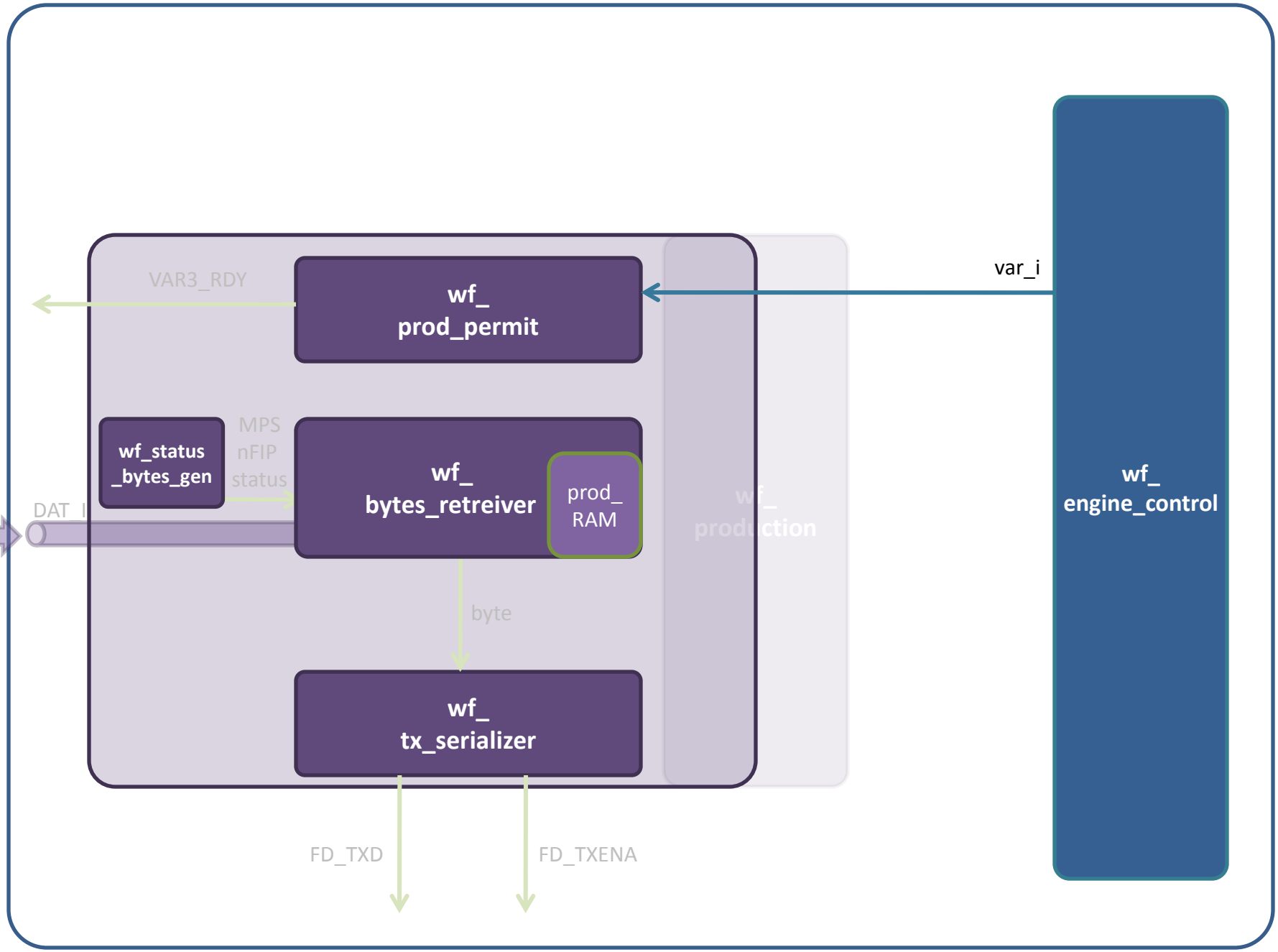

FD_RXD edges
wf_tx_rx_osc
wf_rx_deserializer : PRE detection
nFIP internal or
engine_ctrl
idle
PRE_1st
_f_edge
PRE_
r_edge
r_edge inside the
expected window
PRE_
f_edge
f_edge inside the
expected window
Switch_to
Deglitched
1
0
1
0
osignificant edge windows

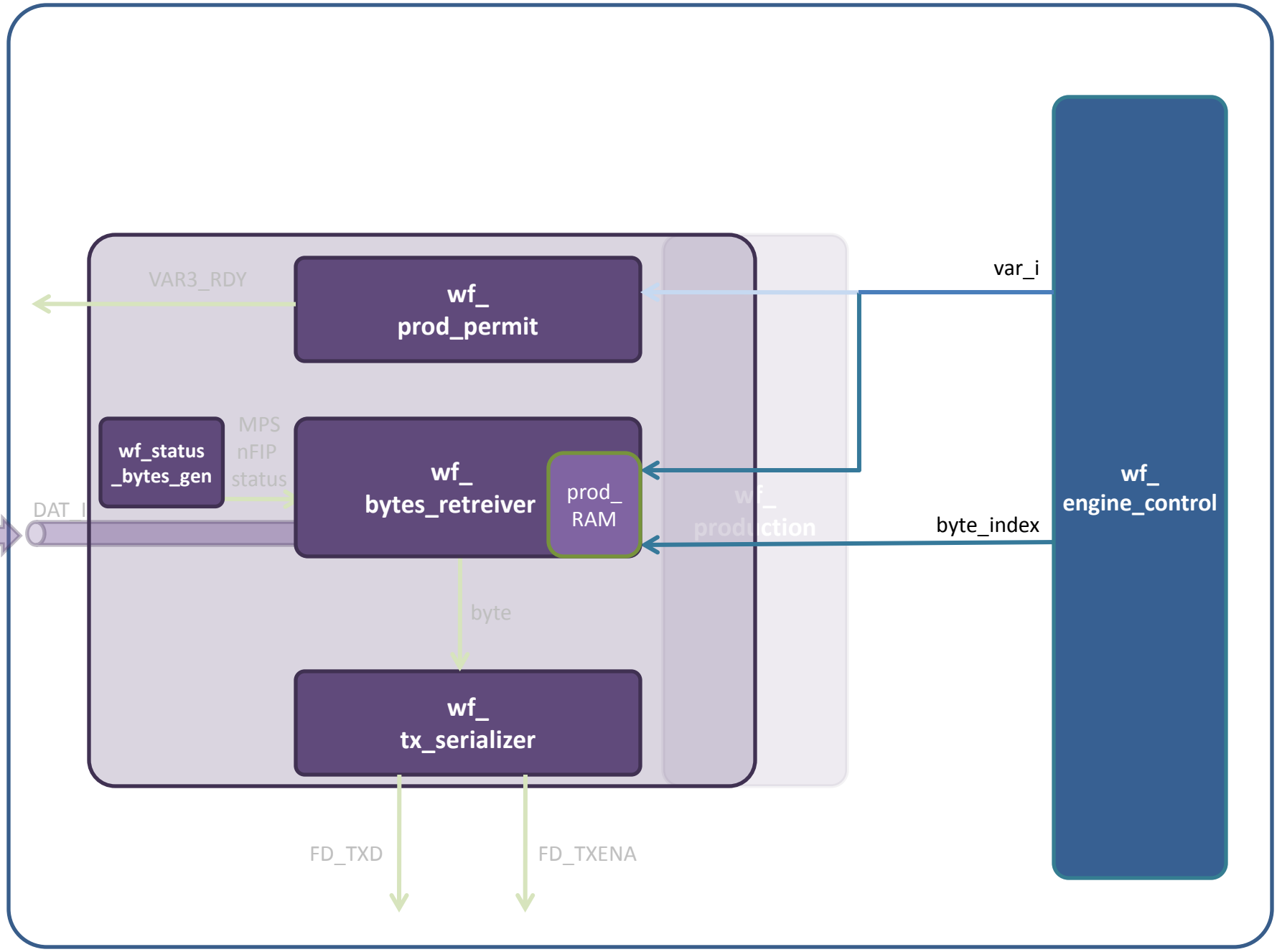

FD_RXD edges
wf_tx_rx_osc
wf_rx_deserializer : rest of frame
idle
wrong bit
FSD
rxd_filtered
sample _manch_bit
FSD last bit
Data_
FCS_
FES
FES OK!
or reset
start CRC
CRC
rxd_filtered
sample_bit
byte
rxd_filtered
sample _manch_bit
FES detection
sample _manch_bit
sample_bit

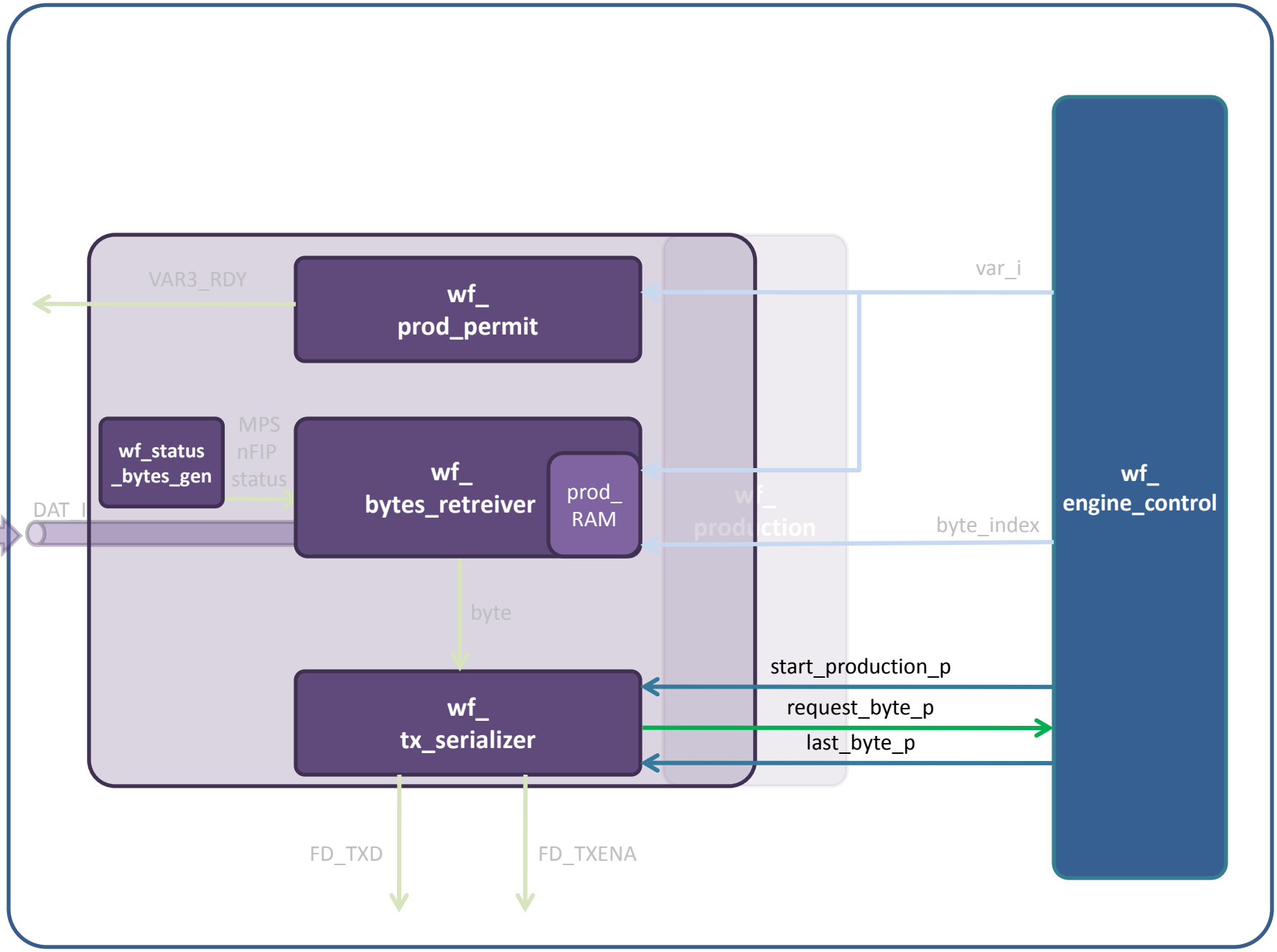

wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

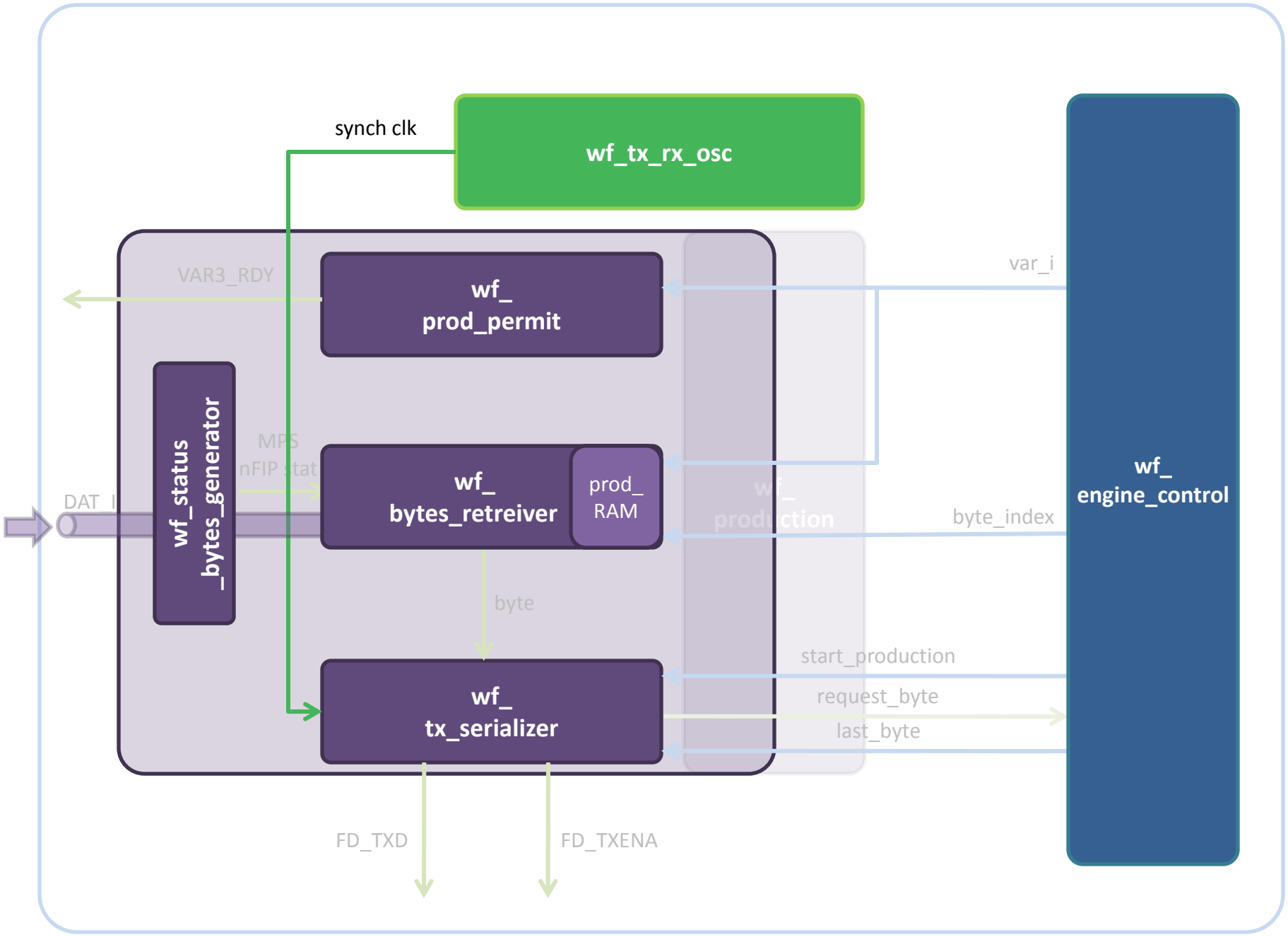

VAR3_RDY
wf_
prod_permit
wf_status
_bytes_gen
wf_
bytes_retreiver
prod_
RAM
wf_
production
DAT_I
wf_
tx_serializer
FD_TXD
FD_TXENA

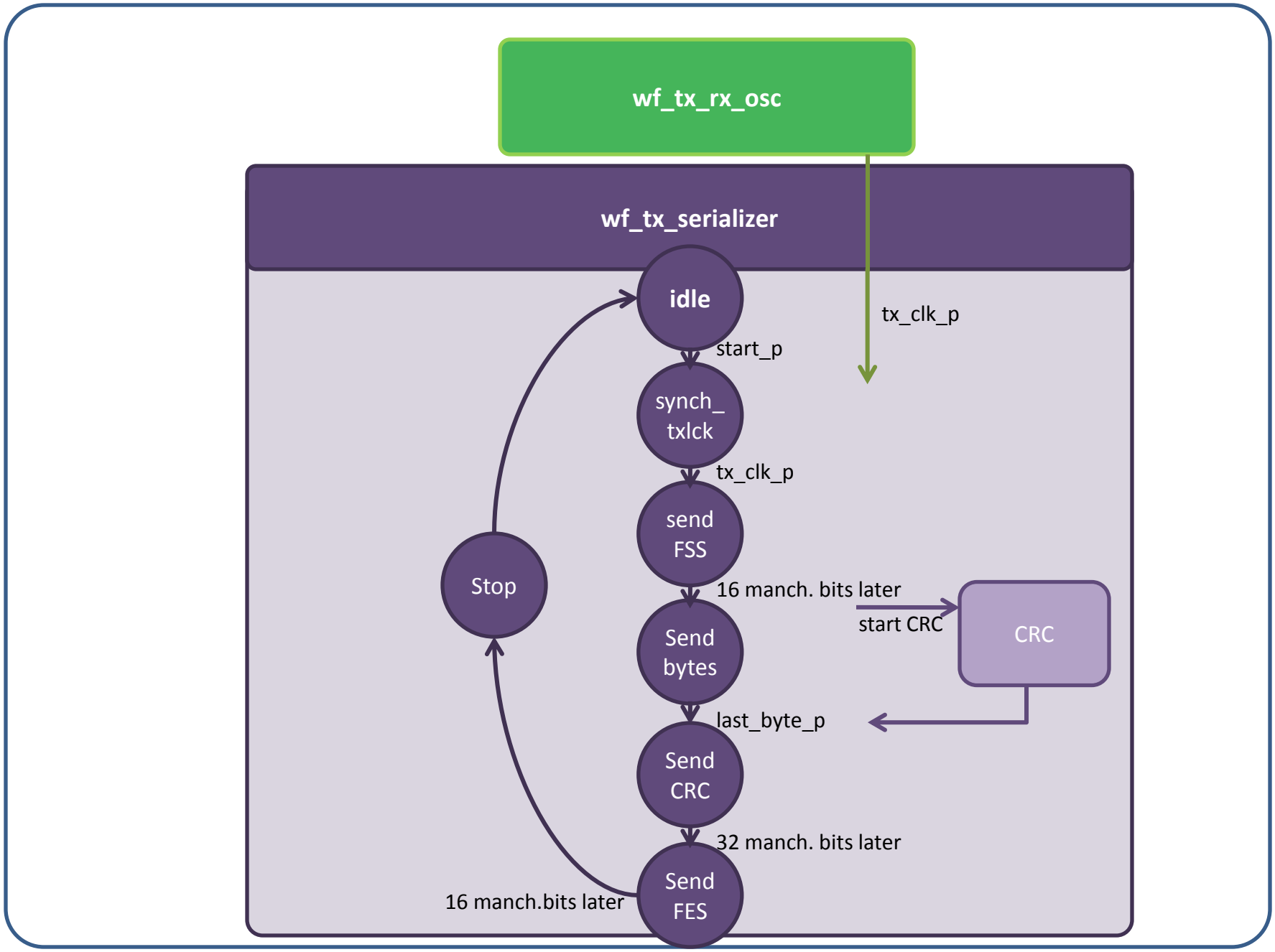

VAR3_RDY
wf_
prod_permit
wf_status
_bytes_gen
MPS
nFIP
status
DAT_I
wf_
bytes_retreiver
prod_
RAM
wf_
production
byte
wf_
tx_serializer
FD_TXD
FD_TXENA

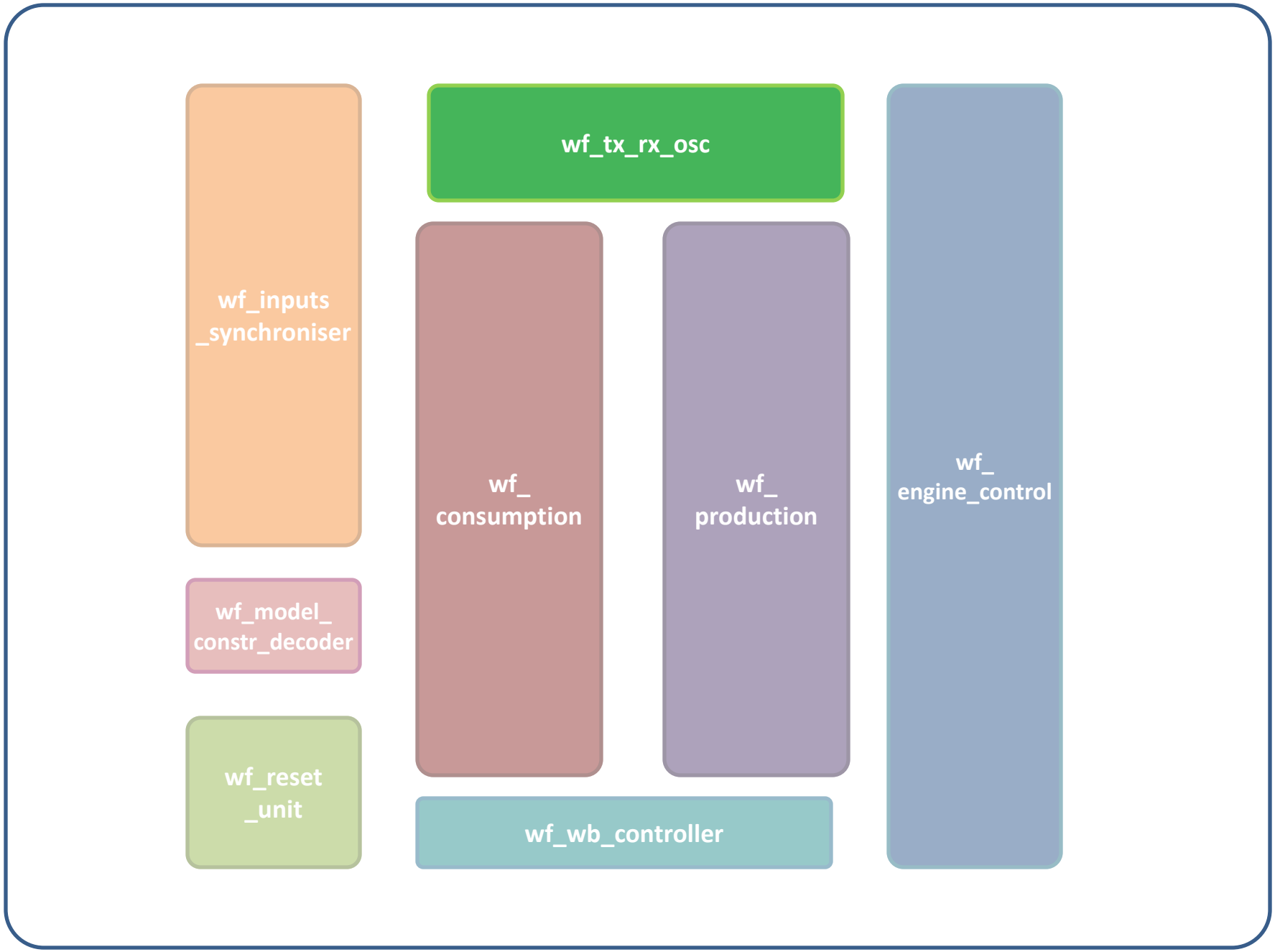

VAR3_RDY
wf_
prod_permit
var_i
wf_status
_bytes_gen
MPS
nFIP
status
wf_
bytes_retreiver
prod_
RAM
wf_
engine_control
DAT_I
byte
wf_
tx_serializer
FD_TXD
FD_TXENA


VAR3_RDY
wf_
prod_permit
var_i
wf_status
_bytes_gen
MPS
nFIP
status
wf_
bytes_retreiver
prod_
RAM
wf_
engine_control
DAT_I
byte_index
byte
wf_
tx_serializer
FD_TXD
FD_TXENA


VAR3_RDY
wf_
prod_permit
var_i
wf_status
_bytes_gen
MPS
nFIP
status
wf_
bytes_retreiver
prod_
RAM
wf_
engine_control
DAT_I
wf_
production
byte_index
byte
start_production_p
request_byte_p
last_byte_p
wf_
tx_serializer
FD_TXD
FD_TXENA


synch clk
wf_tx_rx_osc
VAR3_RDY
wf_
prod_permit
var_i
wf_status
_bytes_generator
MPS
nFIP stat
wf_
bytes_retreiver
prod_
RAM
DAT_I
wf_
production
byte_index
wf_
engine_control
byte
start_production
request_byte
last_byte
wf_
tx_serializer
FD_TXD
FD_TXENA

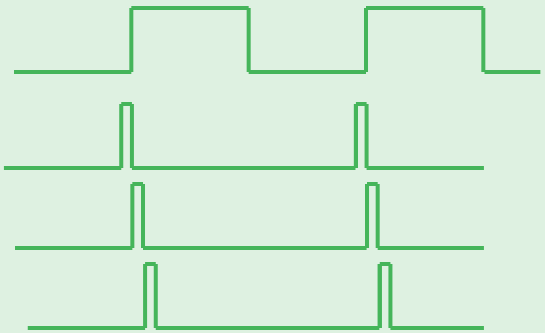

wf_tx_rx_osc
wf_tx_serializer
idle
start_p
synch_
txlck
tx_clk_p
send
FSS
16 manch. bits later
start CRC
CRC
Send
bytes
last_byte_p
Send
CRC
32 manch. bits later
Send
FES
16 manch.bits later
Stop
tx_clk_p


wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

wf_tx_rx_osc
- **tx_osc**

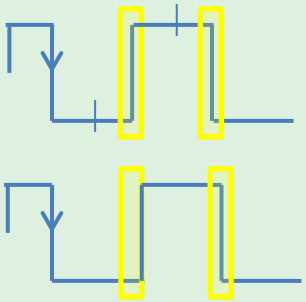
Counter of transmission periods

- FD_TXCK
- tx_clk_p_buff



wf_tx_rx_osc
○ rx_osc

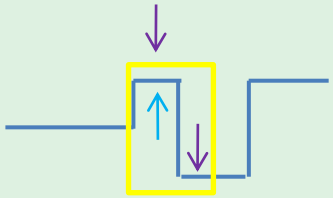
ideally



○ significant edge window

○ adjacent bits transition window

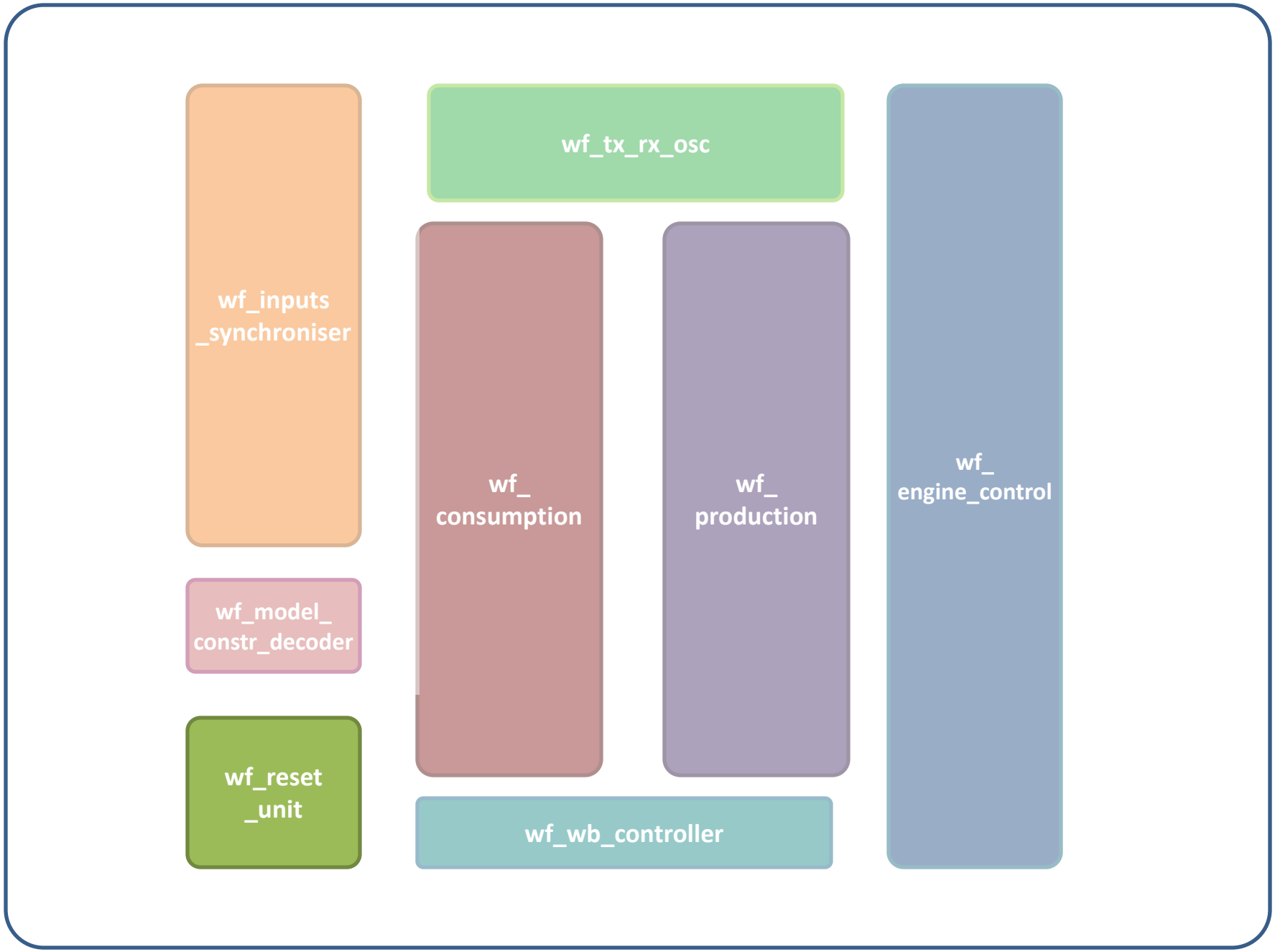
Starts counting after a falling edge on the fd_rxd .
Starting from this edge, other falling or rising significant edges, are expected around one reception period later. A time window around the expected arrival time is set and its length is defined as 1/4th of the period. When the actual edge arrives, the counter is reset.



○ sample_manch_bit_p

○ sample_bit_p

sample_manch_clock: is inverted on each significant edge & between adjacent bits
sample_bit _clock : is inverted only between adjacent bits
The significant edges are normally expected inside the signif_edge_window. In the cases of a code violation (V+ or V-) no edge will arrive in this window. In this situation rx_manch_clk is inverted right after the end of the signif_edge_window.
Edges between adjacent bits are expected inside the adjac_bits_window; if they do not arrive the rx_manch_clk and rx_bit_clk are inverted right after the end of the adjac_bits_window.


wf_inputs
_synchroniser
wf_tx_rx_osc
wf_
consumption
wf_
production
wf_
engine_control
wf_model_
constr_decoder
wf_reset
_unit
wf_wb_controller

nanoFIP reset inputs:

- *Power On Reset*
- User Interface General signal RSTIN
- Reset variable from FIELDRIVE
- Reset to the WISHBONE logic

wf_reset_unit

nanoFIP reset inputs:

- *Power On Reset*
- User Interface General signal RSTIN
- Reset variable from FIELDRIVE
- Reset to the WISHBONE logic

wf_reset _unit

Outputs of this unit :

- nanoFIP internal reset
- FIELDRIVE reset
- user reset RSTON

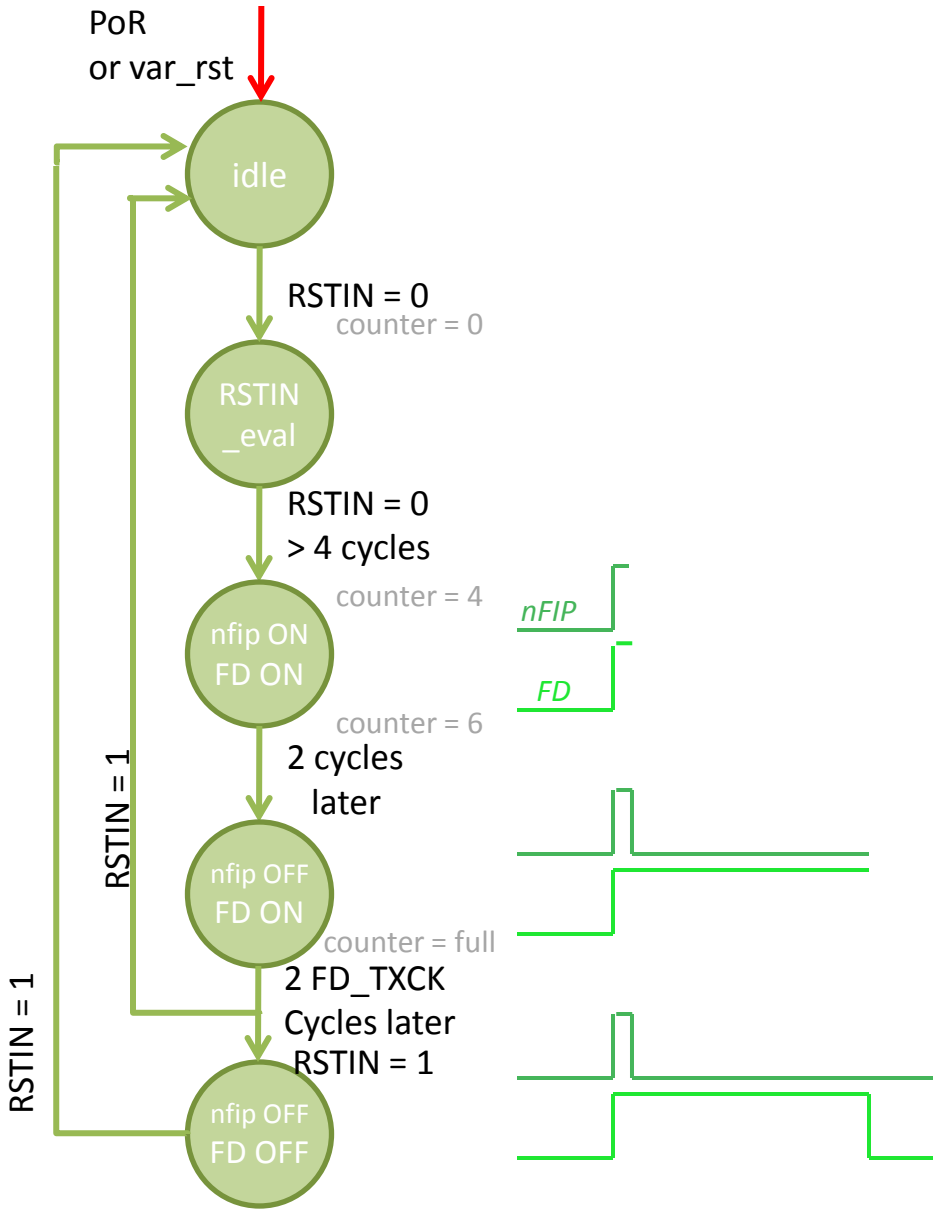
nanoFIP reset inputs:

- *Power On Reset*
- User Interface General signal RSTIN
- Reset variable from FIELDRIVE
- Reset to the WISHBONE logic

wf_reset _unit

Outputs of this unit :

- nanoFIP internal reset
- FIELDRIVE reset
- user reset RSTON



nanoFIP reset inputs:

- *Power On Reset*
- User Interface General signal RSTIN
- Reset variable from FIELDRIVE
- Reset to the WISHBONE logic

wf_reset _unit

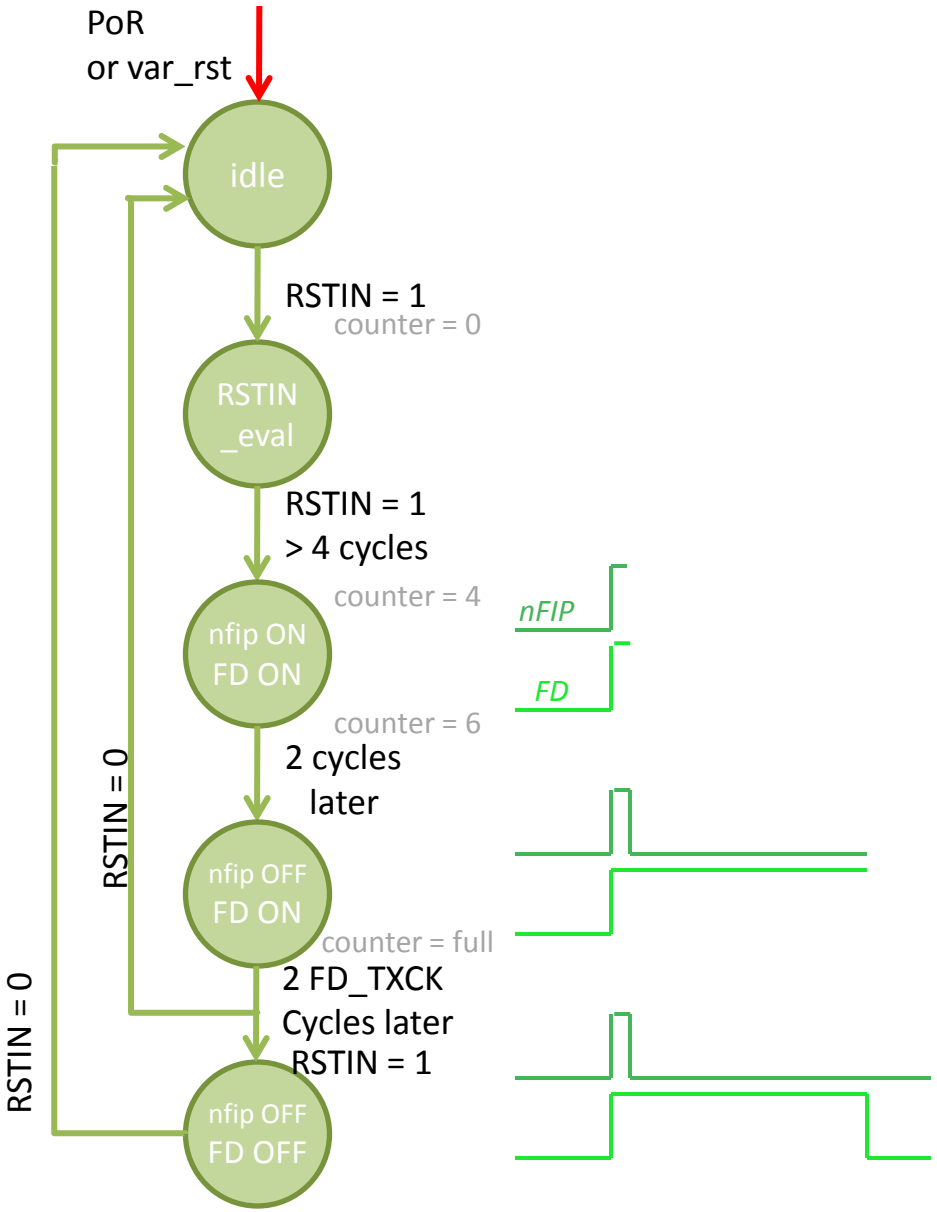
Outputs of this unit :

- nanoFIP internal reset
- FIELDRIVE reset
- user reset RSTON

→ from_RSTIN OR from_var_rst OR from_PoR

→ from_RSTIN OR from_var_rst OR from_PoR

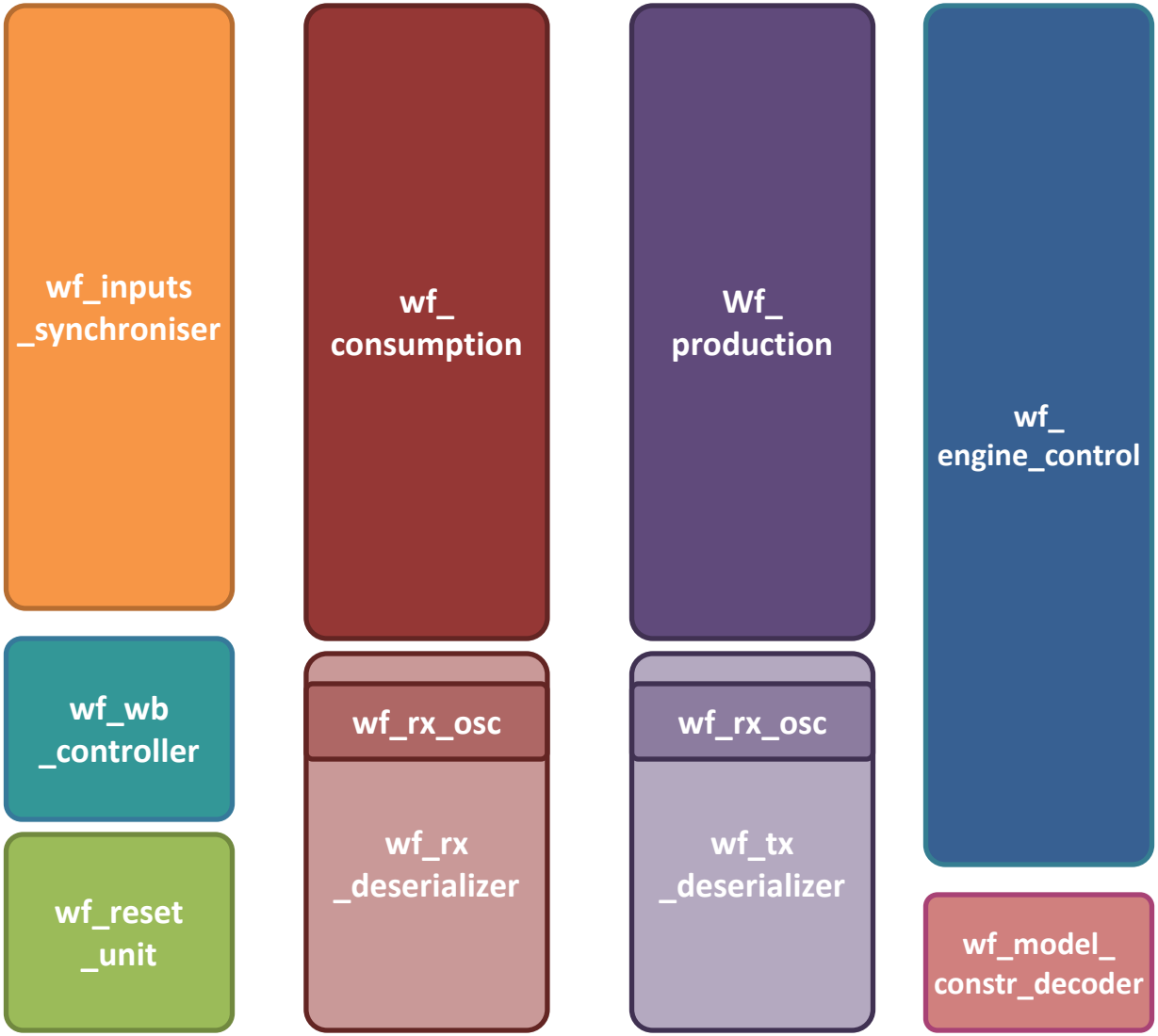
→ from_var_rst



hdl files : http://www.ohwr.org/projects/cern-fip/repository/show/trunk/hdl/design

Actel .prj : http://www.ohwr.org/projects/cern-fip/repository/changes/trunk/hdl/cad/libero/NanoFip/NanoFip.prj

Simulation : http://www.ohwr.org/projects/cern-fip/repository/changes/trunk/software/cadence_IUS/sim.tcl



new nanoFIP main blocks?