

Global Actions

Concurrent Statements

Architecture Declarations

Signals Status

State Register Statements

Process Declarations

Pre Actions:

Post Actions:

Package List

LIBRARY ieee;
 USE ieee.std_logic_1164.all;
 USE ieee.std_logic_arith.all;
 USE ieee.STD_LOGIC_UNSIGNED.all;

SIGNAL	SCOPE	DEFAULT	RESET	STATUS
do	OUT	(OTHERS => '0')	(OTHERS => '0')	CLKD
inc_rescntr	OUT	'0'	'0'	CLKD
inhibit_lls	OUT	'1'	'1'	CLKD
low_rate	OUT	'0'	'0'	CLKD
norm_rate	OUT	'0'	'0'	CLKD
ptc_error	OUT	'0'	'0'	CLKD
ptc_fsm_states	OUT	(OTHERS => '0')	(OTHERS => '0')	CLKD
ptc_status	OUT	X"4"	X"A"	CLKD
ptcstat2led	OUT	B"010"	B"010"	CLKD
res_evnr	OUT	'0'	'0'	CLKD
start_run_fsm	OUT	'0'	'0'	CLKD
stop_run_fsm	OUT	'0'	'0'	CLKD

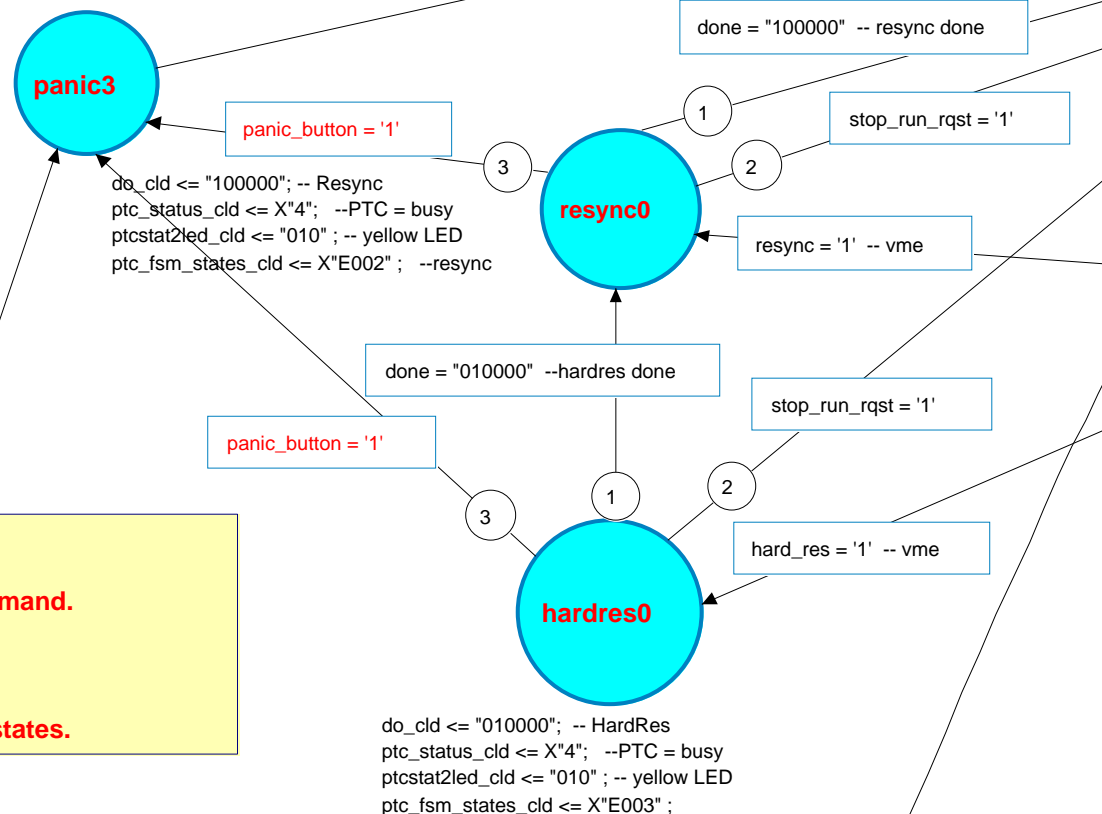
Default values:
 inhibit_l1a = '1'
 ptc_status = X"4" = busy
 ptcstat2led = "010" --yellow
 low_rate = '0'
 norm_rate = '0'

-- BAD CODE is ignored by FSM.

-- do() = long signal...reset by done()
 -- do() as 1Tpulse: delete do command in states.

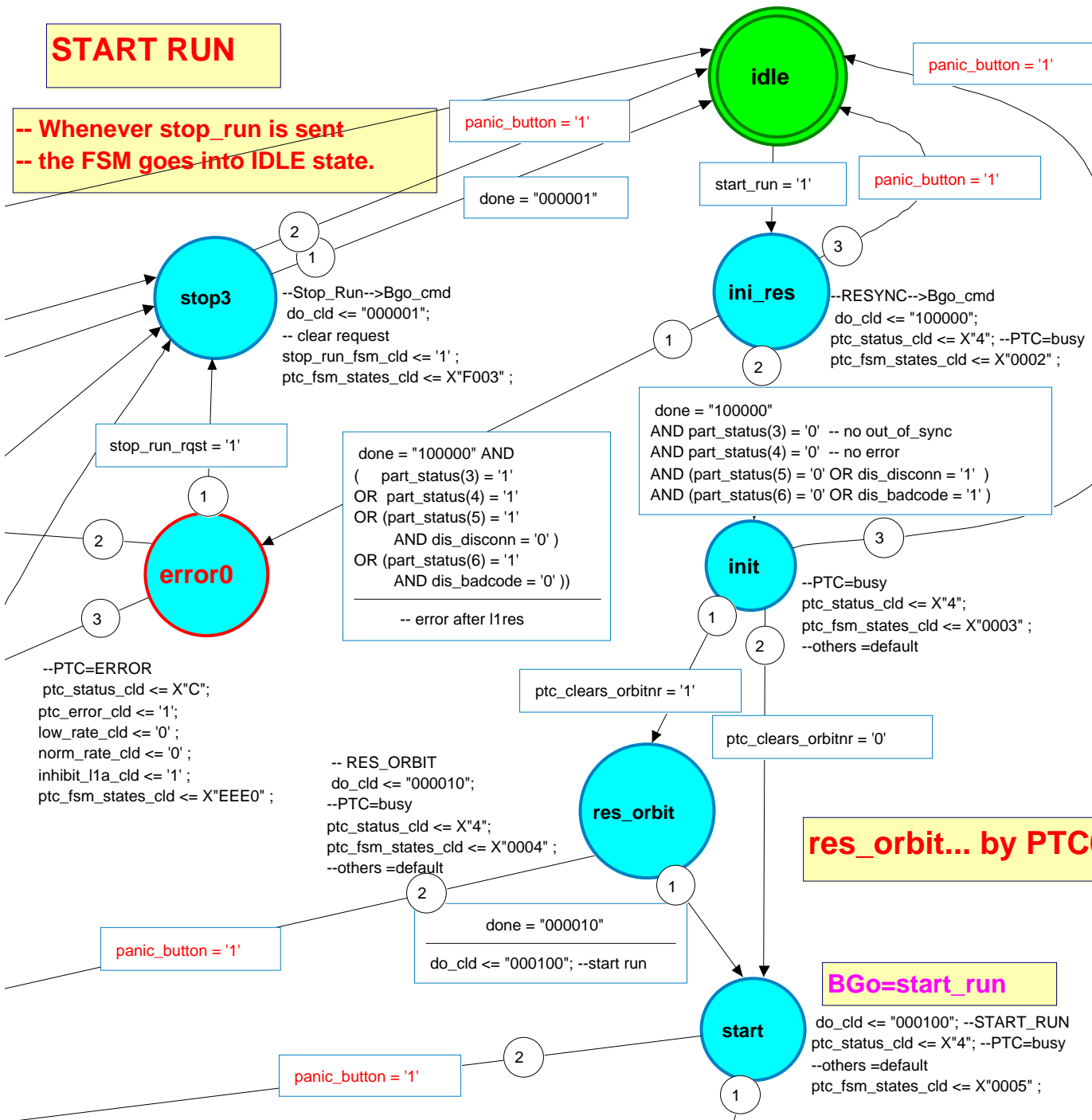
-- V0015: Start independently from input status.
 -- PANIC BUTTON returns ptc_sm to IDLE status if 'done' bit is missing because the BC-table does not contain the bit for this BGo command.
 -- ptc_clears_orbitnr = '1' ..optional reset of orbit counter
 -- IDLE: stop_run_fsm_cld <= '1'; ' ignore old stop request
 -- 'hard_res' is possible when in out_of_sync status
 -- V0011: resync_vme, hardres_vme possible when in busy, warn, ready states.

Vienna		Project:	tcs_chip
		V0015	
Title:	PTC State Machine		
Path:	tcs_chip_lib/ptc_sm/fsm		
Edited:	by taurok on 21 Nov 2007		



START RUN

-- Whenever stop_run is sent
-- the FSM goes into IDLE state.



```

do_cld <= (others => '0'); -- default
ptc_error_cld <= '0'; -- default
ptc_status_cld <= X"A"; -- IDLE ...default
inhibit_l1a_cld <= '1'; -- default
low_rate_cld <= '0'; -- default
norm_rate_cld <= '0'; -- default
res_evr_cld <= '0'; -- default
ptcstat2led_cld <= "010"; -- default
ptc_fsm_states_cld <= X"0001";
-- clear old stop rqst from a bad end of run
stop_run_fsm_cld <= '1';
    
```

panic_button = old 'res_ptc'

- do <= 000 001 = StopRun
- do <= 000 010 = ResOrbit
- do <= 000 100 = Start Run
- do <= 001 000 = ResEvrnr
- do <= 010 000 = Hardres
- do <= 100 000 = Resync

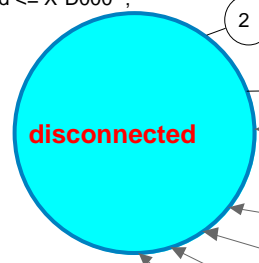
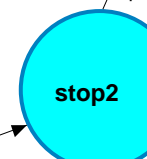
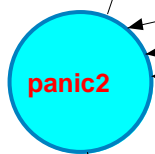
-- Input
-- PARTITION STATES
-- part_status :=decoded bits
bit 0: READY
bit 1: WARNING
bit 2: BUSY
bit 3: OUT_OF_SYNC
bit 4: ERROR
bit 5: DISCONNECTED
bit 6: BAD CODE
bit 7: ---free

-- Output
-- PTC STATUS --> DAQ
-- 4 bits encoded
0000=X"0": OFF
0001=X"1": WARNING
0010=X"2": OUT_OF_SYNC
0100=X"4": BUSY
1000=X"8": READY
1010=X"A": IDLE
1100=X"C": ERROR

res_orbit... by PTC0 or by VME only

BGo=start_run

-- FRONT PANEL LEDs
-- ptcstat2led(2) = ready --> green LED
-- ptcstat2led(1) = idle, busy, warn --> yellow LED (default)
-- ptcstat2led(0) = error, resync --> red LED



```
--PTC=waiting for detector  
ptc_status_cld <= X"0"; --disconnected input  
ptc_error_cld <= '0';  
low_rate_cld <= '0';  
norm_rate_cld <= '0';  
inhibit_l1a_cld <= '1';  
ptc_fsm_states_cld <= X"D000";
```

ptc_fsm_states_cld <= X"F002";

stop_run_rqst = '1'

-- connected again
part_status(5) = '0' OR dis_disconn = '1'

-- disconnected
part_status(5) = '1' AND dis_disconn = '0'

-- disconnected
part_status(5) = '1' AND dis_disconn = '0'

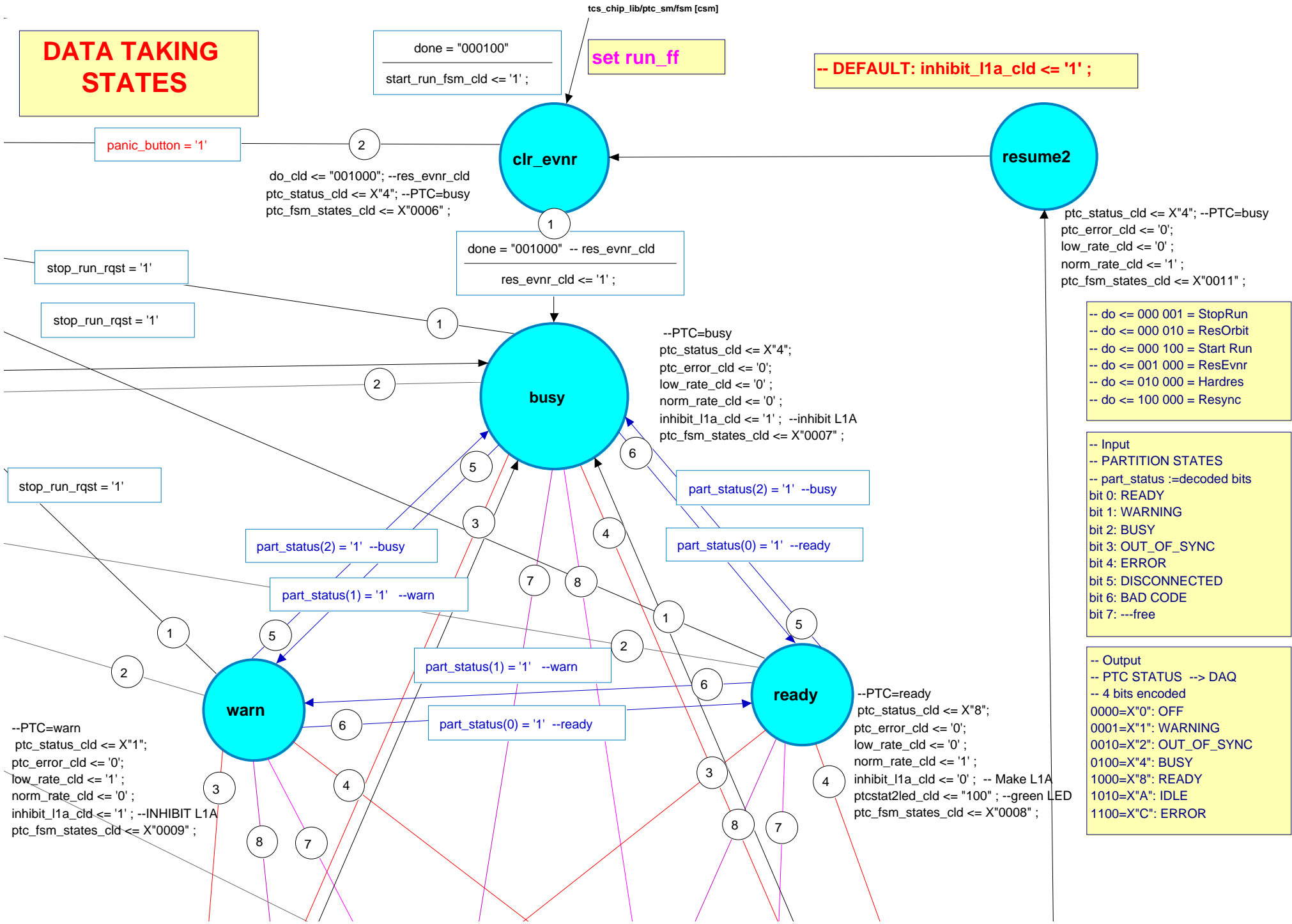
-- disconnected
part_status(5) = '1' AND dis_disconn = '0'

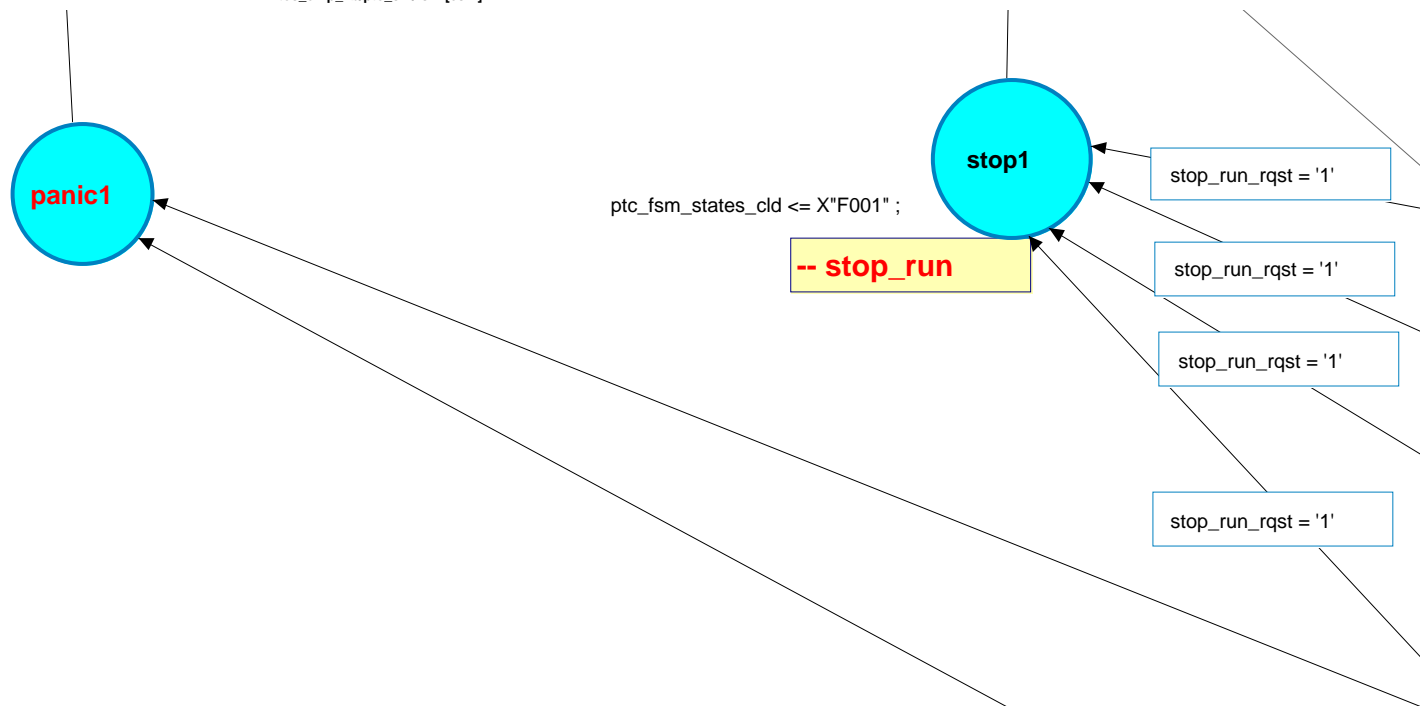
-- disconnected
part_status(5) = '1' AND dis_disconn = '0'

-- disconnected
part_status(5) = '1' AND dis_disconn = '0'

-- PTC waits until disconnected Detector Partition has been reconnected to resume.

DATA TAKING STATES





ptc_fsm_states_cld <= X"F001" ;

-- stop_run

```

-- do <= 000 001 = StopRun
-- do <= 000 010 = ResOrbit
-- do <= 000 100 = Start Run
-- do <= 001 000 = ResEvr
-- do <= 010 000 = Hardres
-- do <= 100 000 = Resync
    
```

```

-- Input
-- PARTITION STATES
-- part_status :=decoded bits
bit 0: READY
bit 1: WARNING
bit 2: BUSY
bit 3: OUT_OF_SYNC
bit 4: ERROR
bit 5: DISCONNECTED
bit 6: BAD CODE
bit 7: ---free
    
```

