SMP @ MPP

Stephane GABOURIN

Benjamin TODD

Ivan ROMERA

Amanda GARCIA

Maciej KWIATKOWSKI

Nikolai TROFIMOV

Maxime AUDRAIN

Menu

Introduction	B. Todd
Hardware	M. Kwiatkowski
Testing and Testers	S. Gabourin
Software	I. Romera
User Interface	M. Audrain
Status & Future	I. Romera

SMP 3v0
Introduction

Safe Machine Parameters

receives accelerator information

generates flags & values

directly transmitted and / or broadcast

injection procedure

Extraction Interlocks

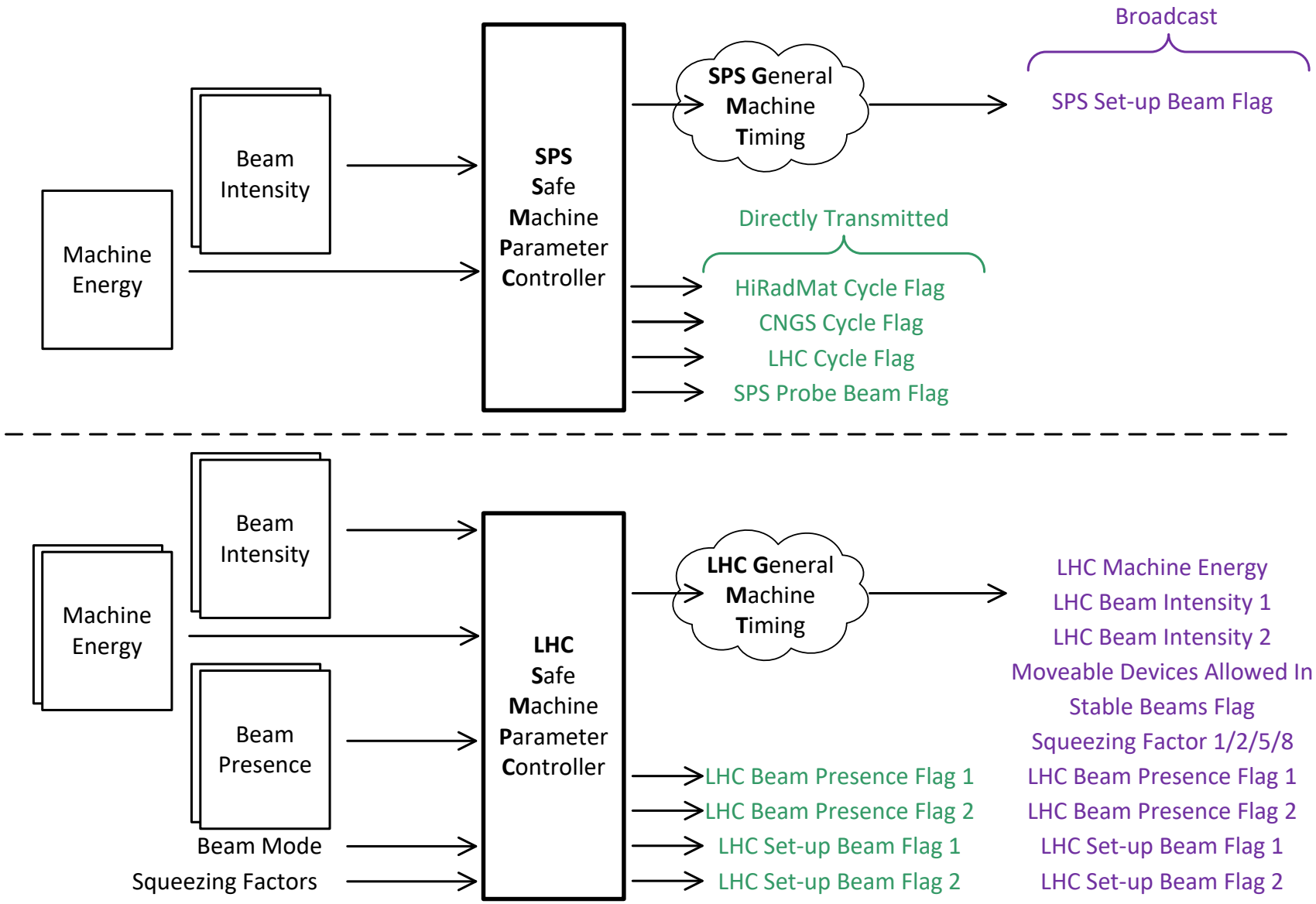
protection configuration

Beam Interlocks
Collimation
Beam Loss Monitors …

*fast *safe *reliable *available

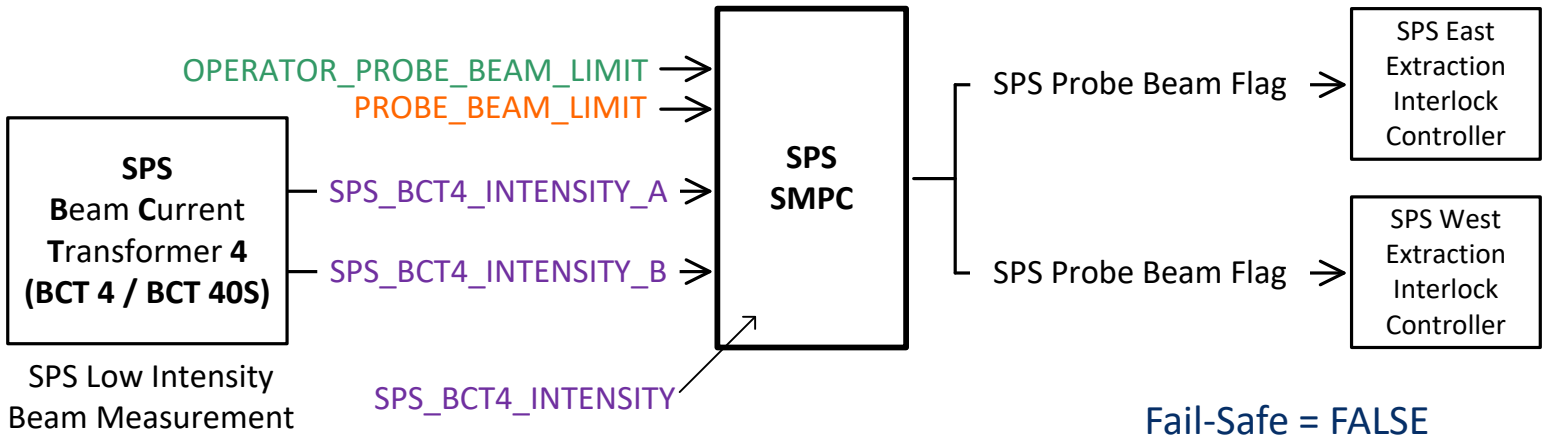
CERN = System Safety

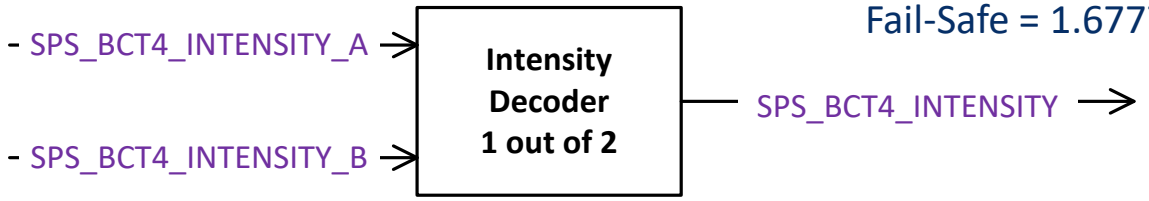
Two Controllers



SPS
Parameters

SPS Probe Beam Flag



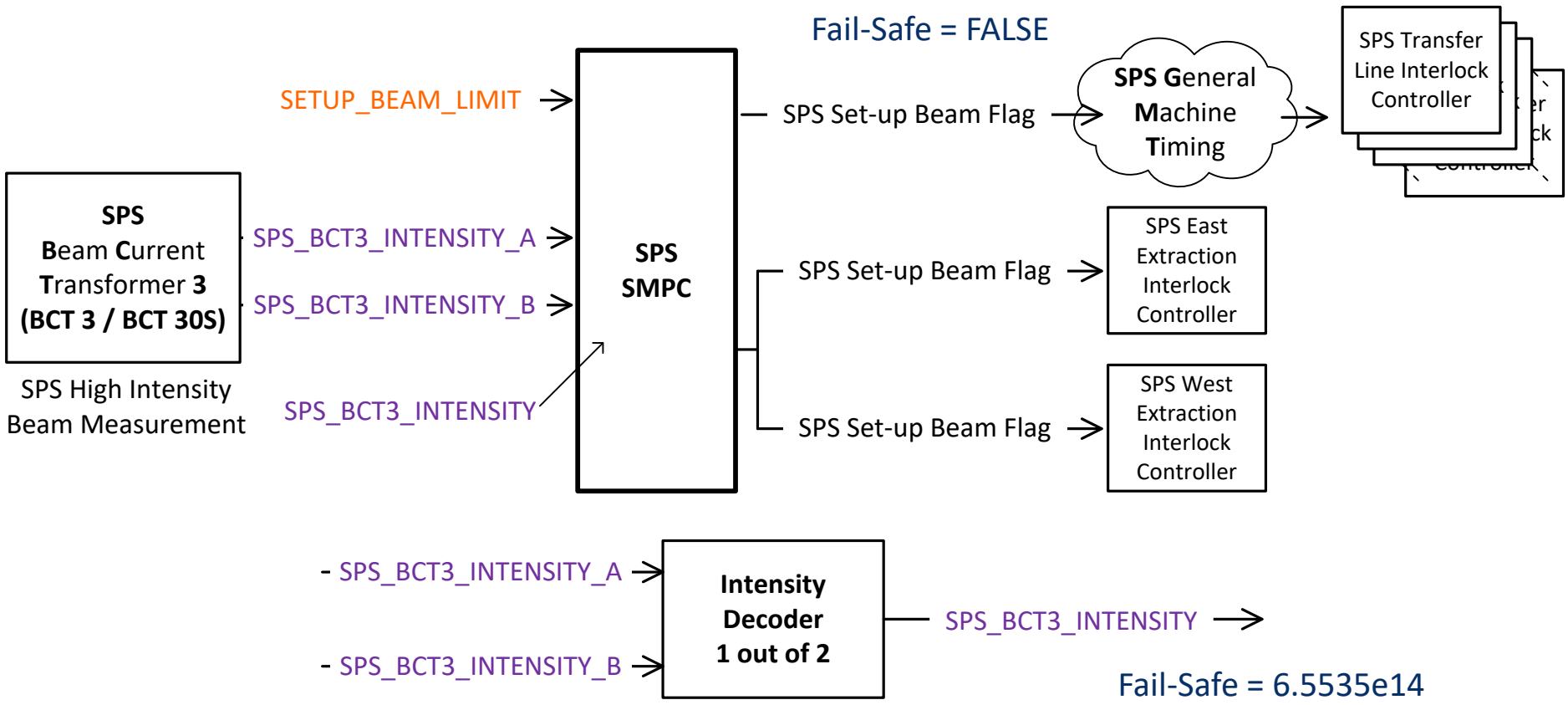


SPS_PBF = TRUE when
(SPS_BCT4_INTENSITY ≤ PROBE_BEAM_LIMIT)
else SPS_PBF = FALSE

PROBE_BEAM_LIMIT may be trimmed down by OPERATOR_PROBE_BEAM_LIMIT

OPERATOR_PROBE_BEAM_LIMIT = 1.4e10
PROBE_BEAM_LIMIT = 1.0e11

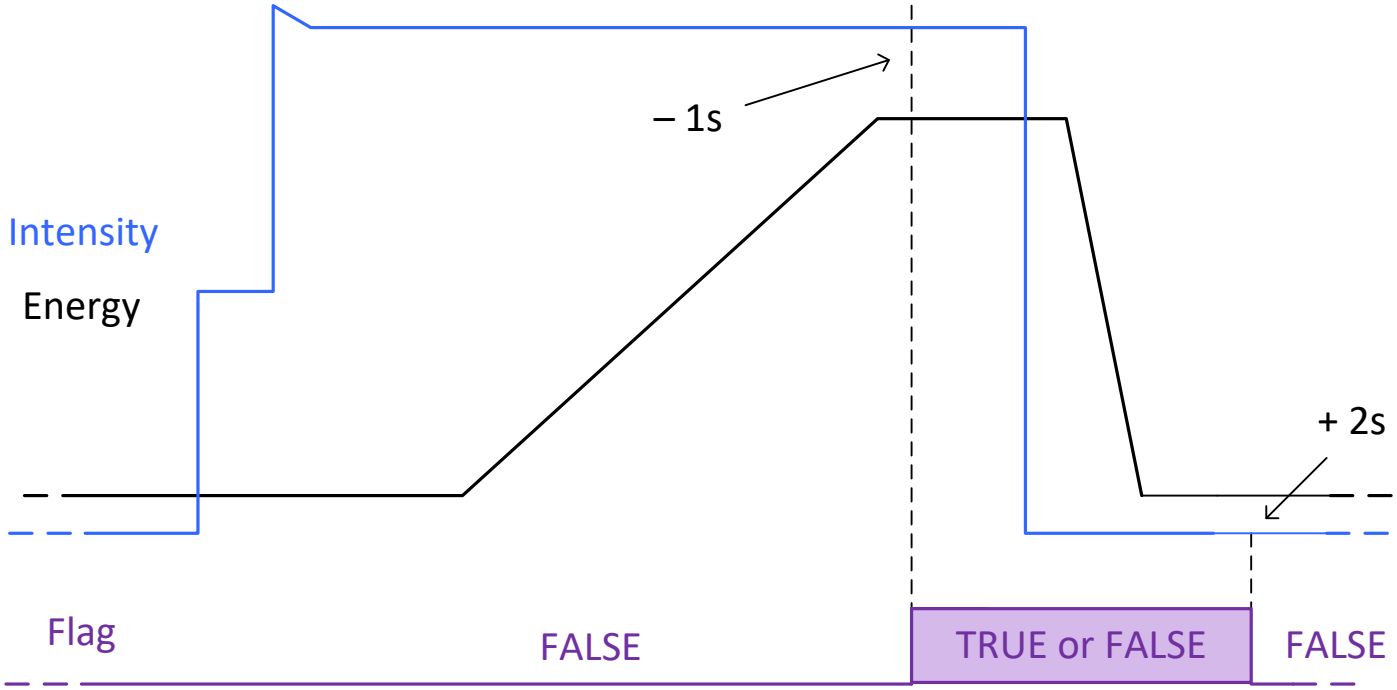
SPS Setup Beam Flag



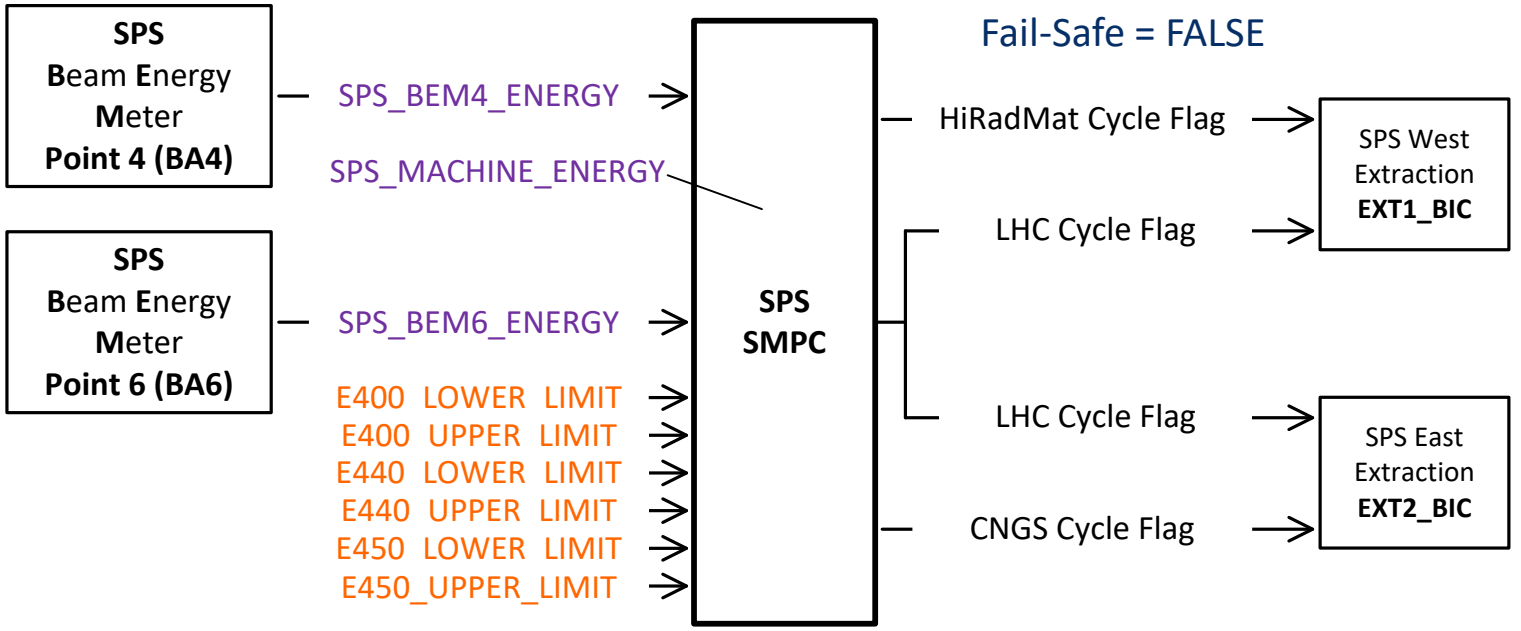
SPS_SBF = TRUE when
(SPS_BCT3_INTENSITY ≤ SETUP_BEAM_LIMIT)
else SPS_SBF = FALSE

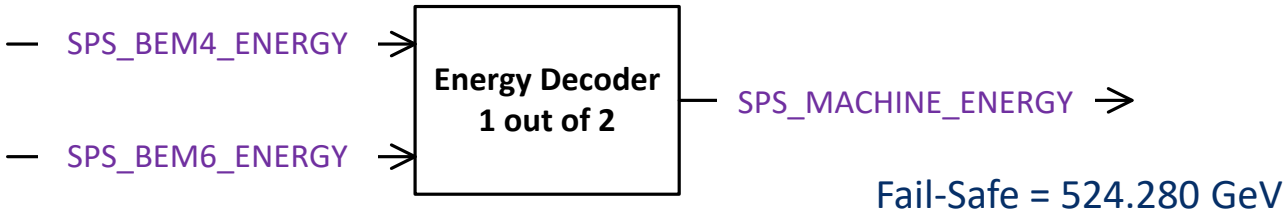
SETUP_BEAM_LIMIT = 5.0e11

Probe / Setup Timing



SPS Energy Flags



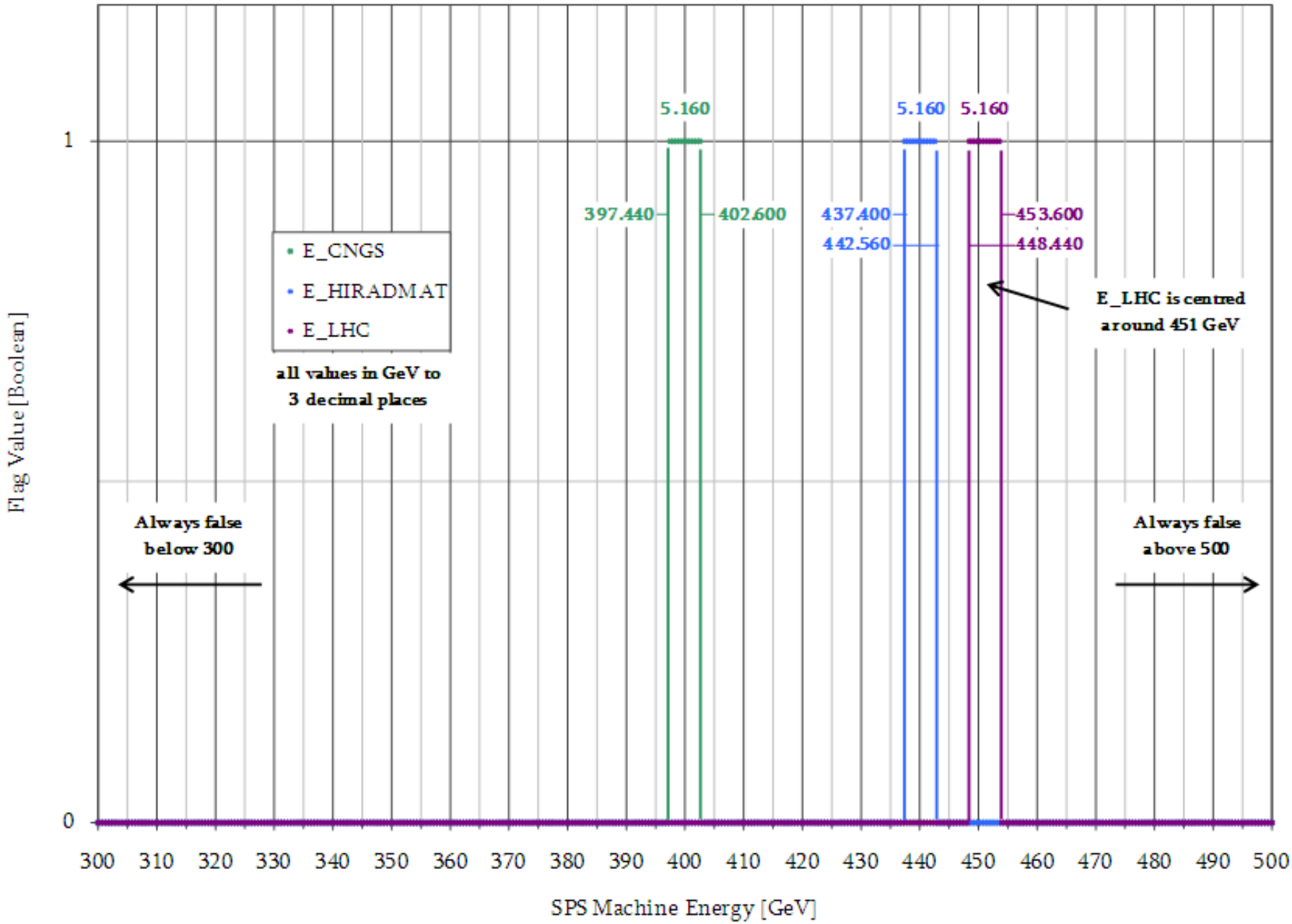


SPS Energy Flags

E_CNGS = TRUE when
[SPS_MACHINE_ENERGY ≥ E400_LOWER_LIMIT = 397.440 GeV
AND
SPS_MACHINE_ENERGY ≤ E400_UPPER_LIMIT] = 402.600 GeV
else E_CNGS = FALSE

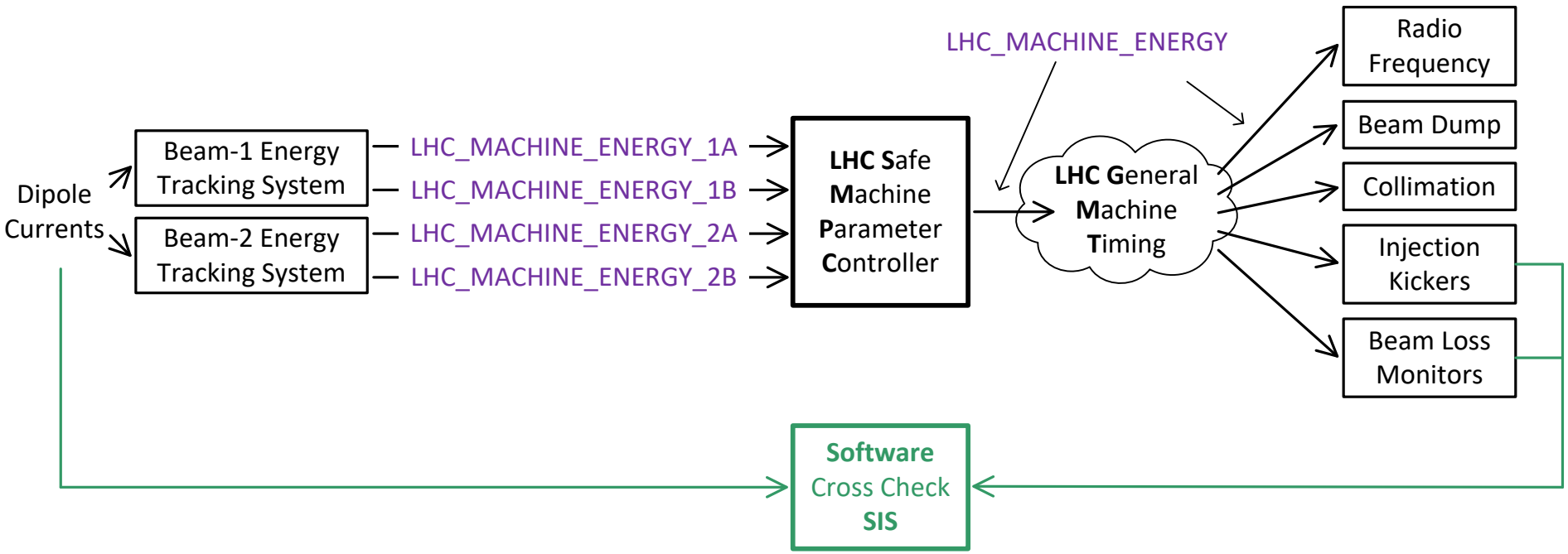
E_ HIRADMAT = TRUE when
[SPS_MACHINE_ENERGY ≥ E440_LOWER_LIMIT = 437.400 GeV
AND
SPS_MACHINE_ENERGY ≤ E440_UPPER_LIMIT] = 442.560 GeV
else E_LHC = FALSE

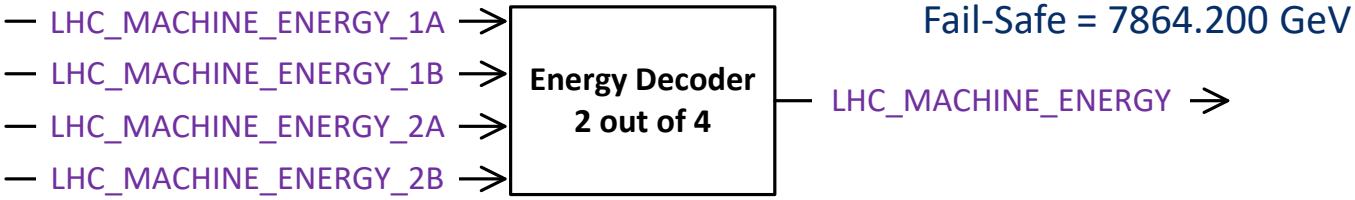
E_LHC = TRUE when
[SPS_MACHINE_ENERGY ≥ E450_LOWER_LIMIT = 448.440 GeV
AND
SPS_MACHINE_ENERGY ≤ E450_UPPER_LIMIT] = 453.600 GeV
else E_LHC = FALSE


5.160
5.160
5.160
1
397.440
402.600
437.400
442.560
453.600
448.440
E_CNGS
E_HIRADMAT
E_LHC
all values in GeV to
3 decimal places
E_LHC is centred
around 451 GeV
Always false
below 300
Always false
above 500
Flag Value [Boolean]
0
300 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 480 490 500
SPS Machine Energy [GeV]

LHC
Parameters

LHC Energy

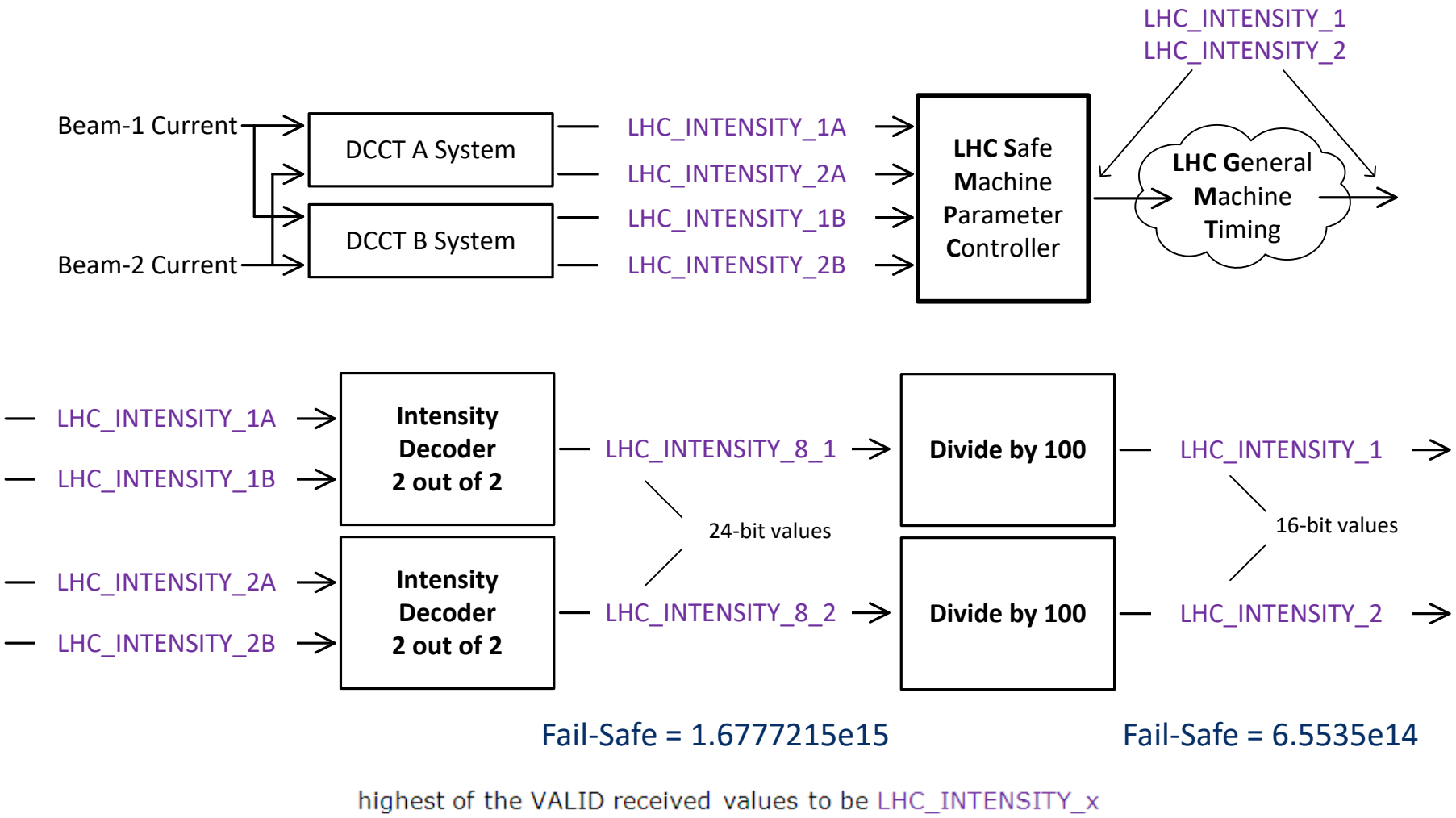




Fail-Safe = 7864.200 GeV

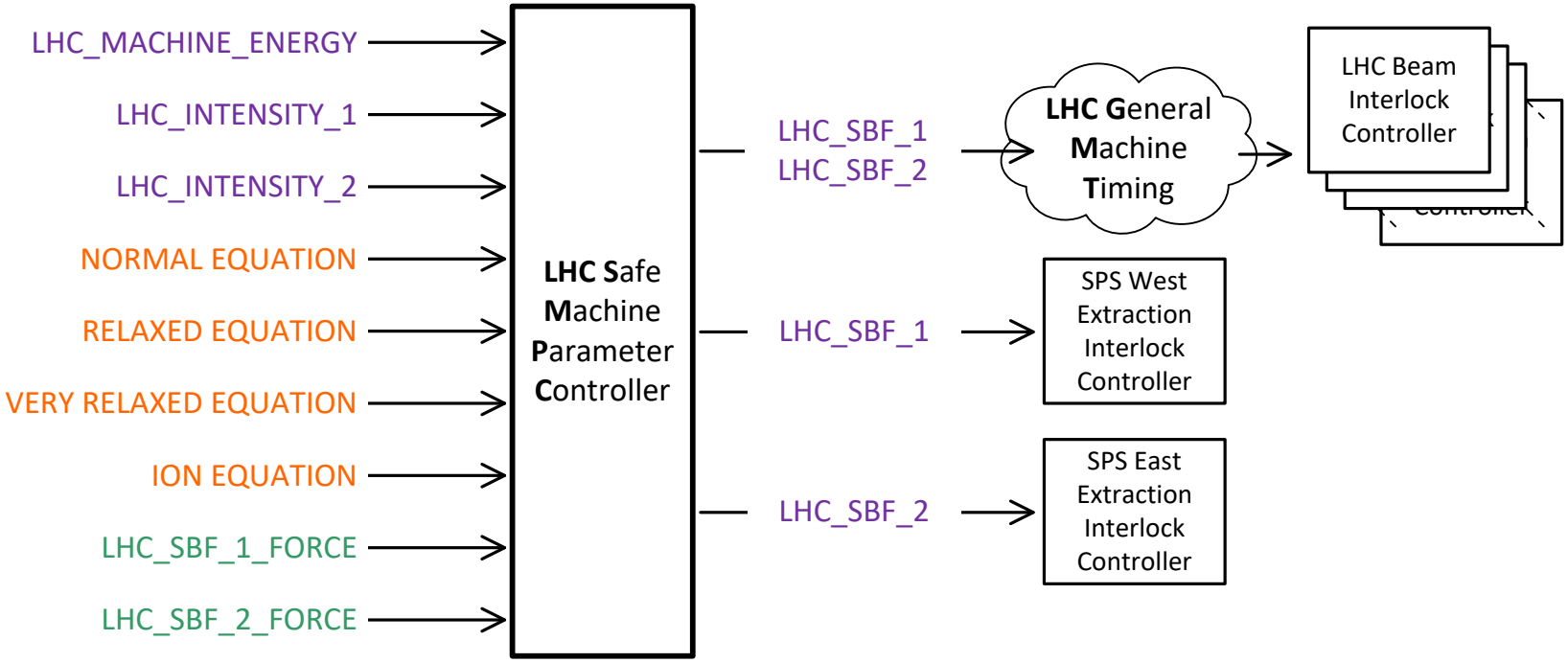
highest of the VALID received values to be LHC_MACHINE_ENERGY

LHC Intensity

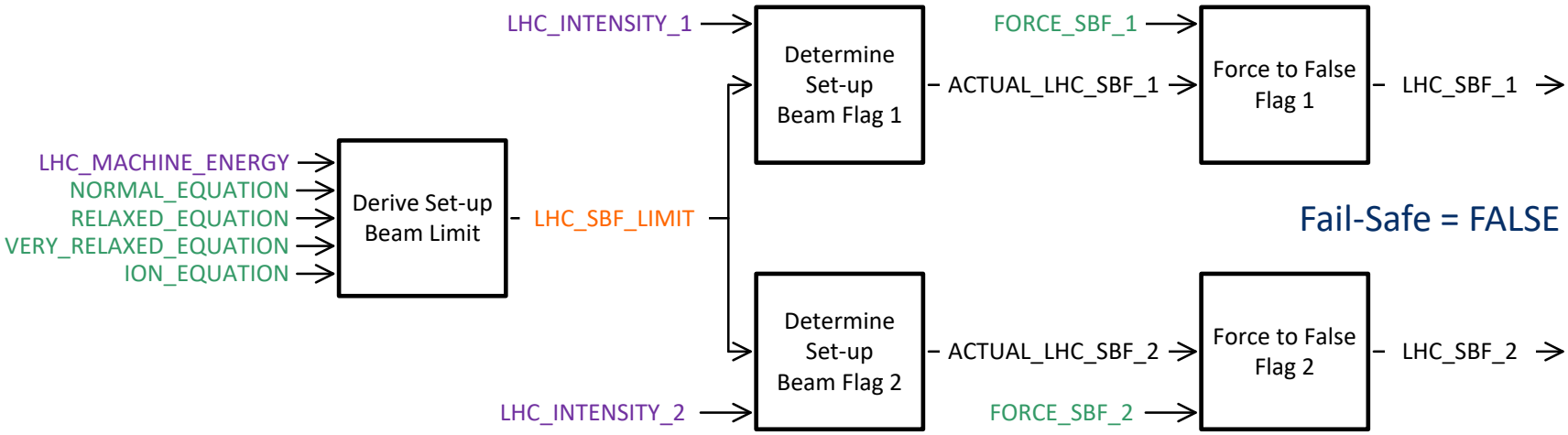


highest of the VALID received values to be LHC_INTENSITY_x

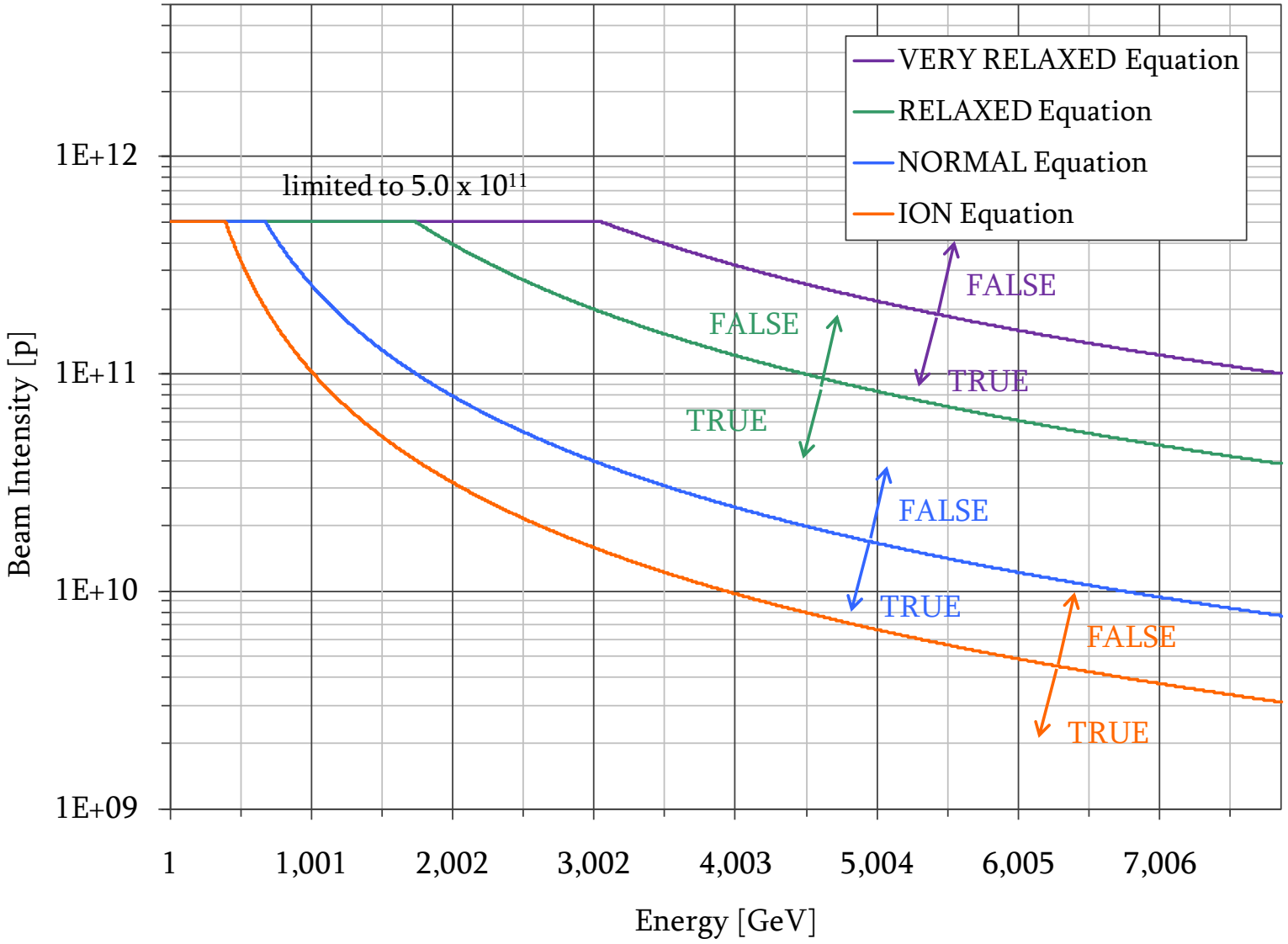
LHC Set-up Beam Flag



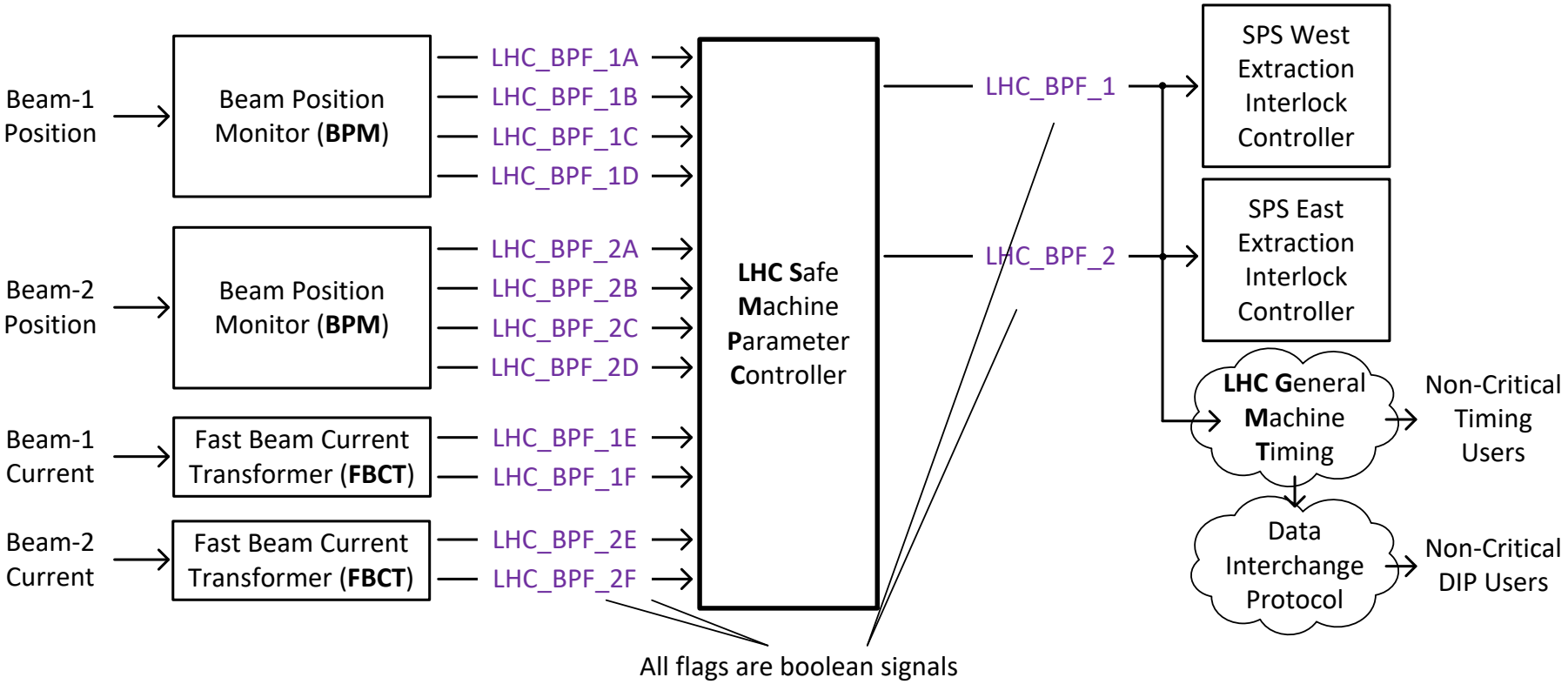
LHC Set-up Beam Flag



LHC Set-up Beam Flag

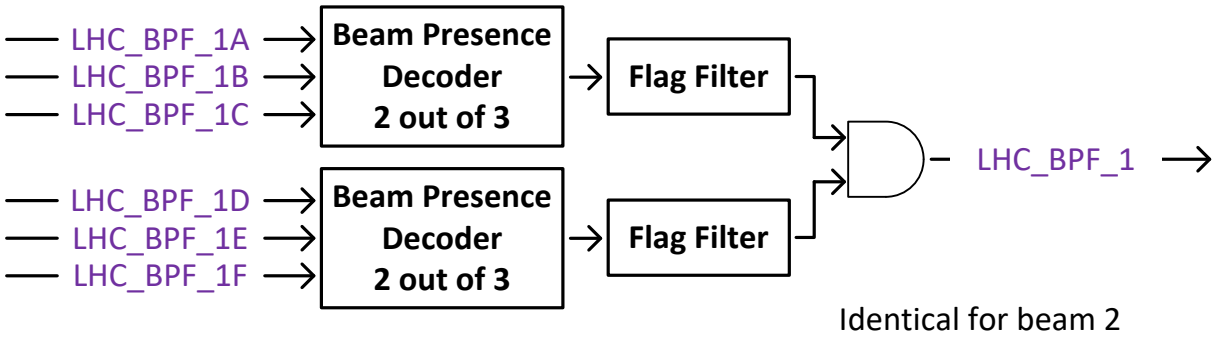


LHC Beam Presence Flag



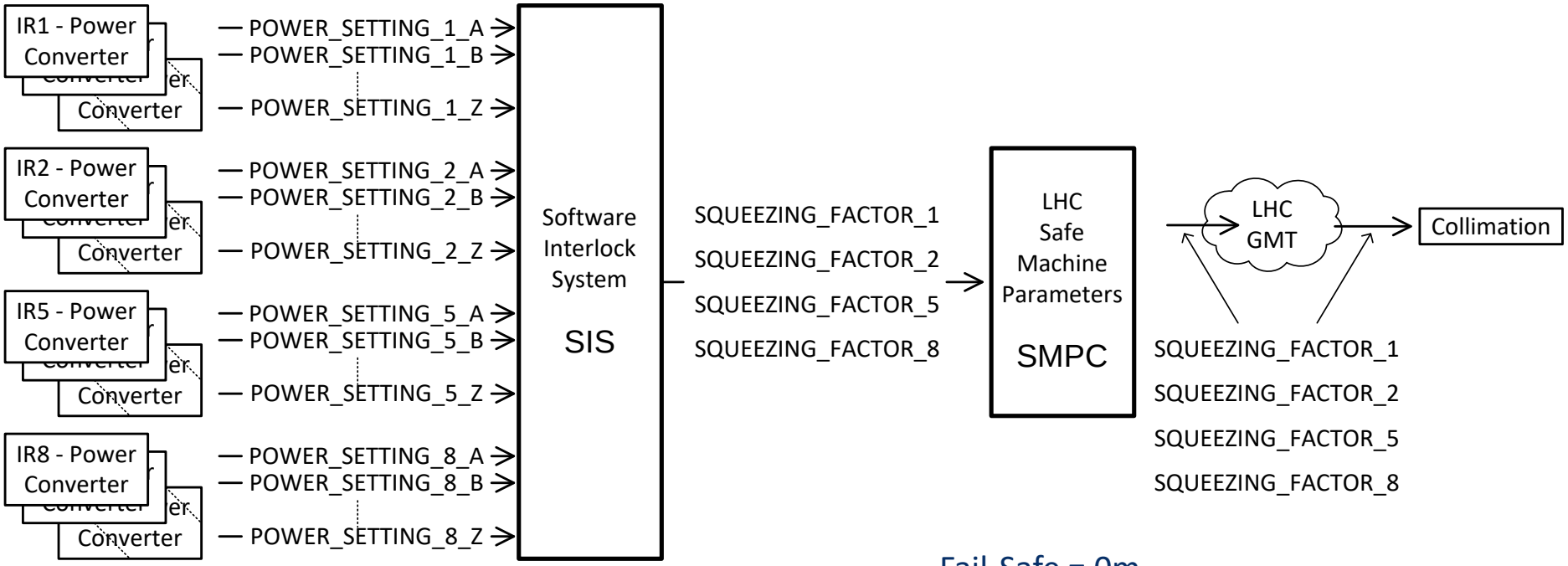
LHC Beam Presence Flag

Fail-Safe = FALSE



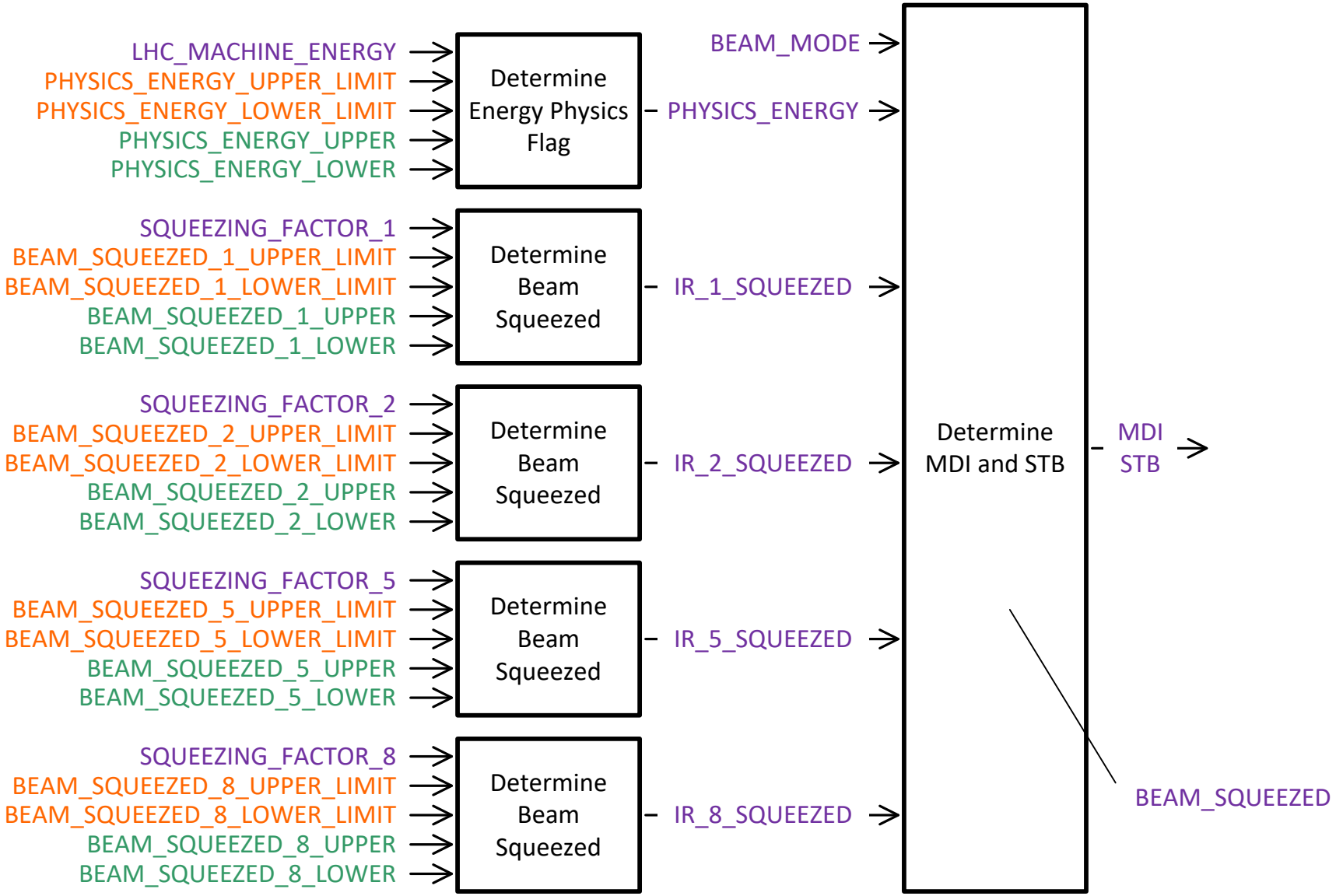
Identical for beam 2

Squeezing Factors

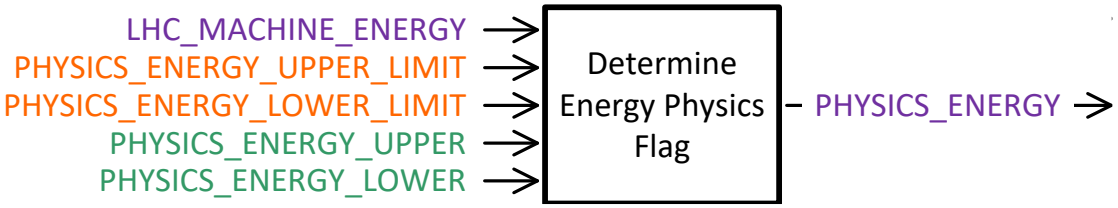


Fail-Safe = 0m

Moveable Devices and Stable Beams



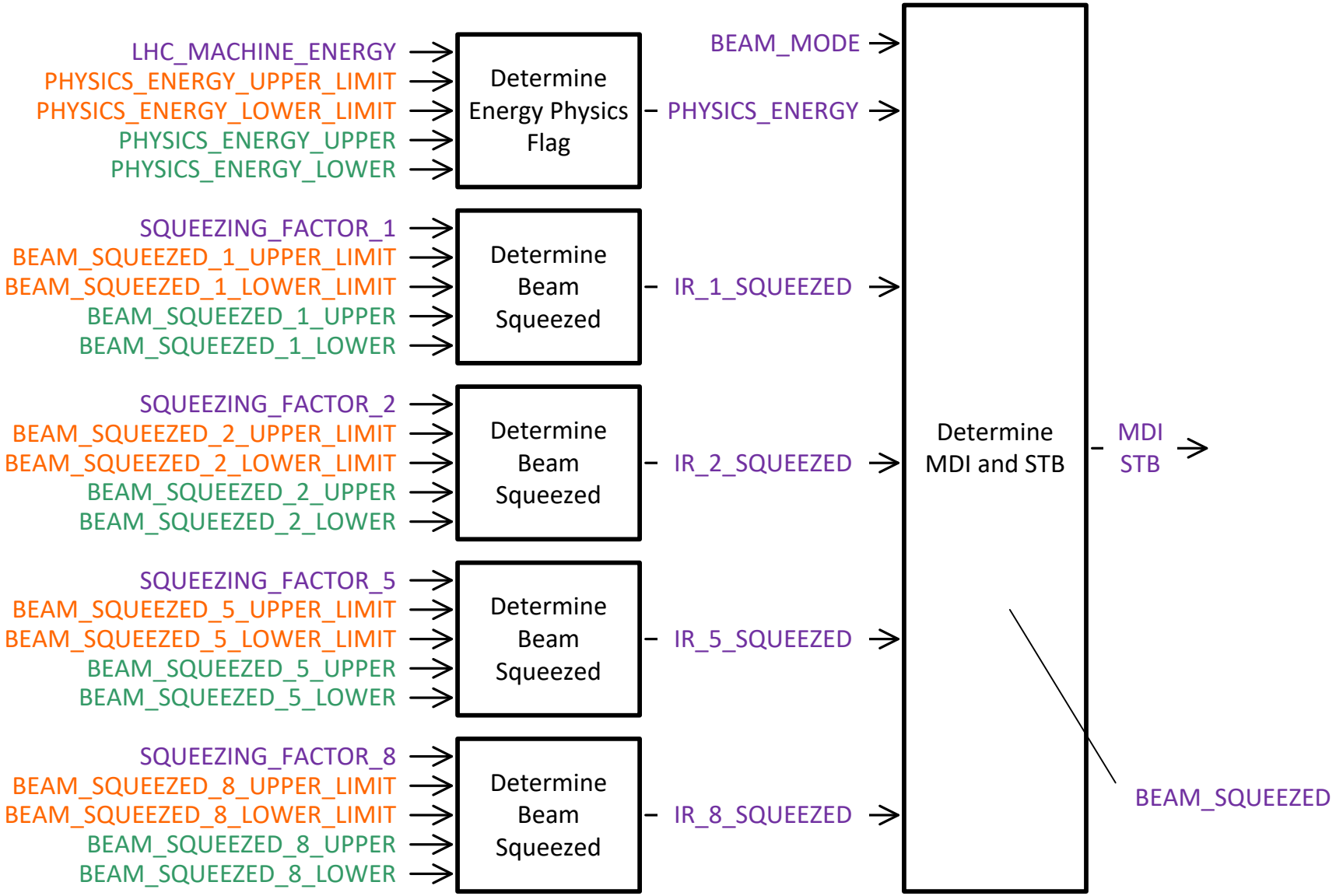
Moveable Devices and Stable Beams



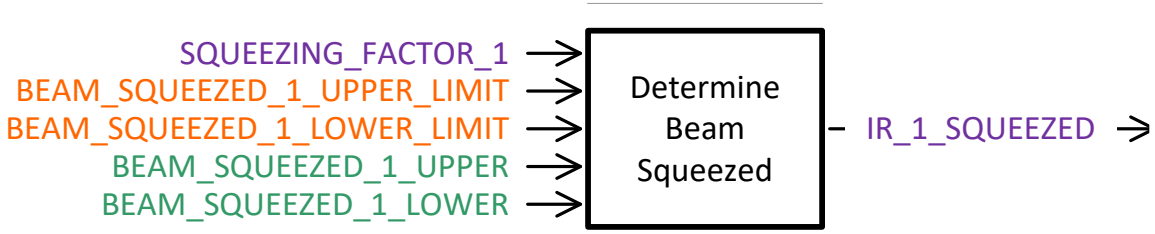
if (PHYSICS_ENERGY_LOWER ≤ LHC_MACHINE_ENERGY ≤ PHYSICS_ENERGY_UPPER)
PHYSICS_ENERGY = TRUE
else PHYSICS_ENERGY = FALSE

Operator: <10 GeV window between LIMITs

Moveable Devices and Stable Beams



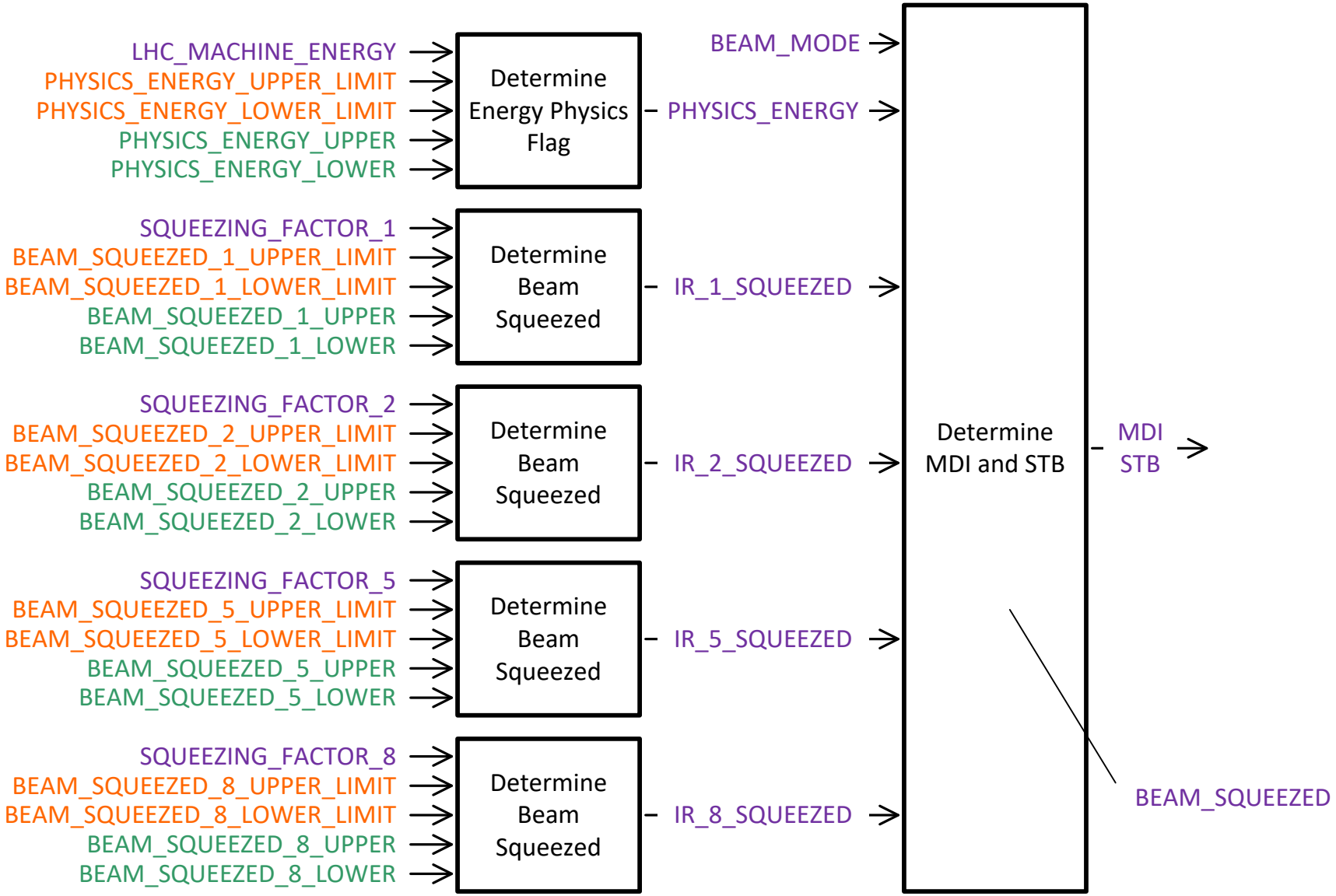
Moveable Devices and Stable Beams



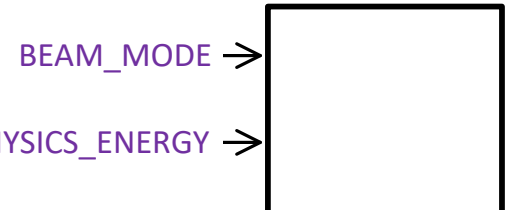
if (BEAM_SQUEEZED_n_ LOWER ≤ SQUEEZING_FACTOR_n ≤ BEAM_SQUEEZED_n_UPPER)
IR_n_SQUEEZED = TRUE
else IR_n_SQUEEZED = FALSE

Operator: <1m window between LIMITs

Moveable Devices and Stable Beams



Moveable Devices and Stable Beams



MDI Flag calculation:

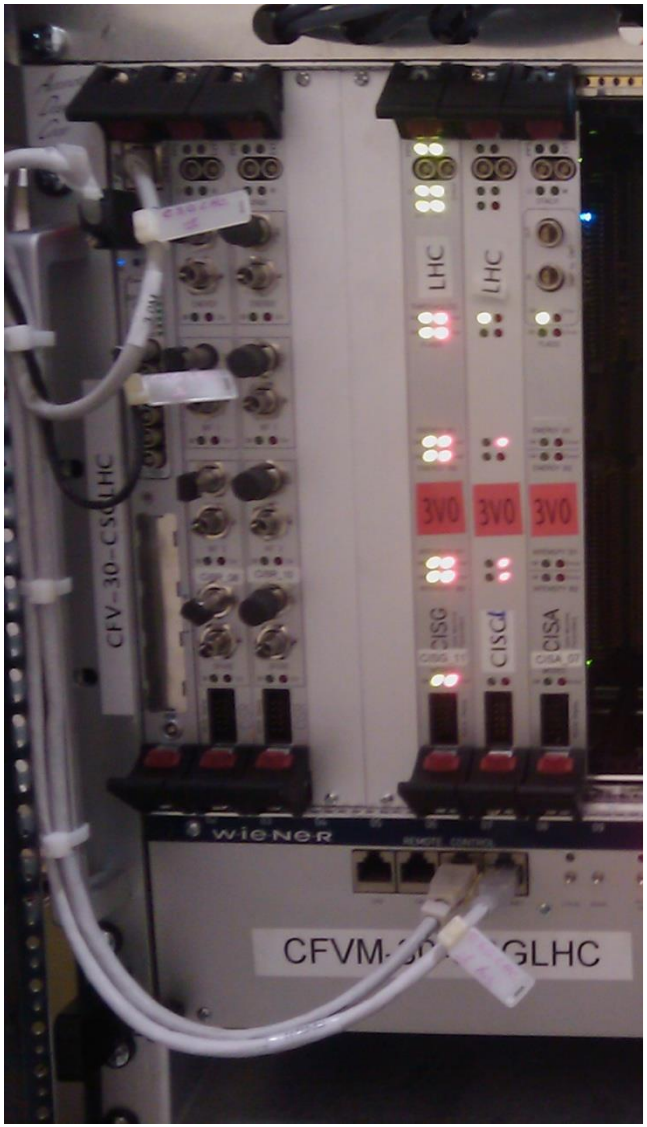
if (PHYSICS_ENERGY = TRUE)
AND
(BEAM_MODE = "STABLE BEAMS" OR BEAM_MODE = "UNSTABLE BEAMS" OR
BEAM_MODE = "BEAM DUMP")
AND
(BEAM_SQUEEZED = TRUE) then
MDI = TRUE
Else MDI = FALSE

STB Flag calculation:

if (PHYSICS_ENERGY = TRUE)
AND
(BEAM_MODE = "STABLE BEAMS")
AND
(BEAM_SQUEEZED = TRUE) then
STB = TRUE
Else STB = FALSE

Dependable Electronics
Basis

VME Chassis & Generic Circuit - CISX

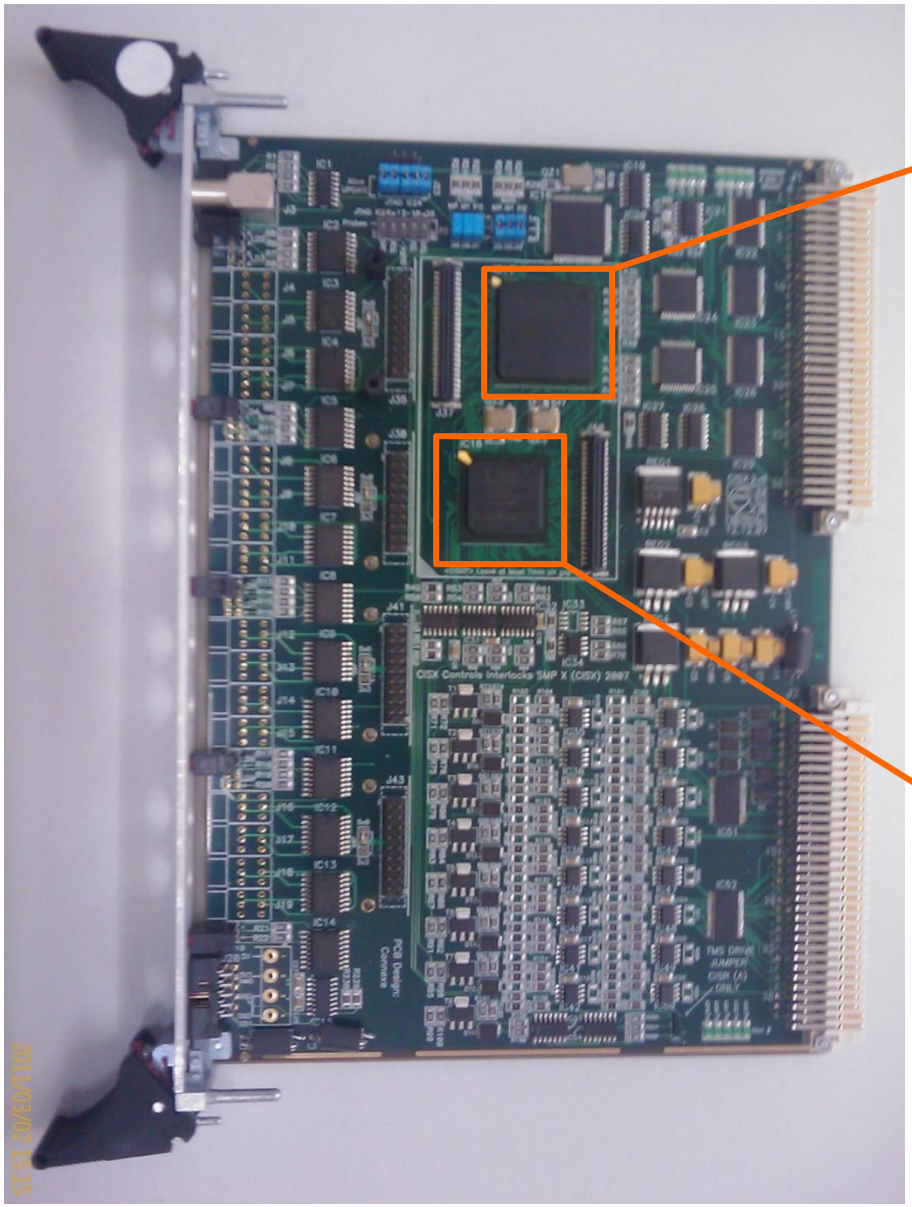


Receiver

Generator LHC
or
Generator SPS

Arbiter

VME Chassis & Generic Circuit - CISX



Monitor FPGA

Receiver – CISR

Generator LHC – CISGL

Generator SPS – CISGS

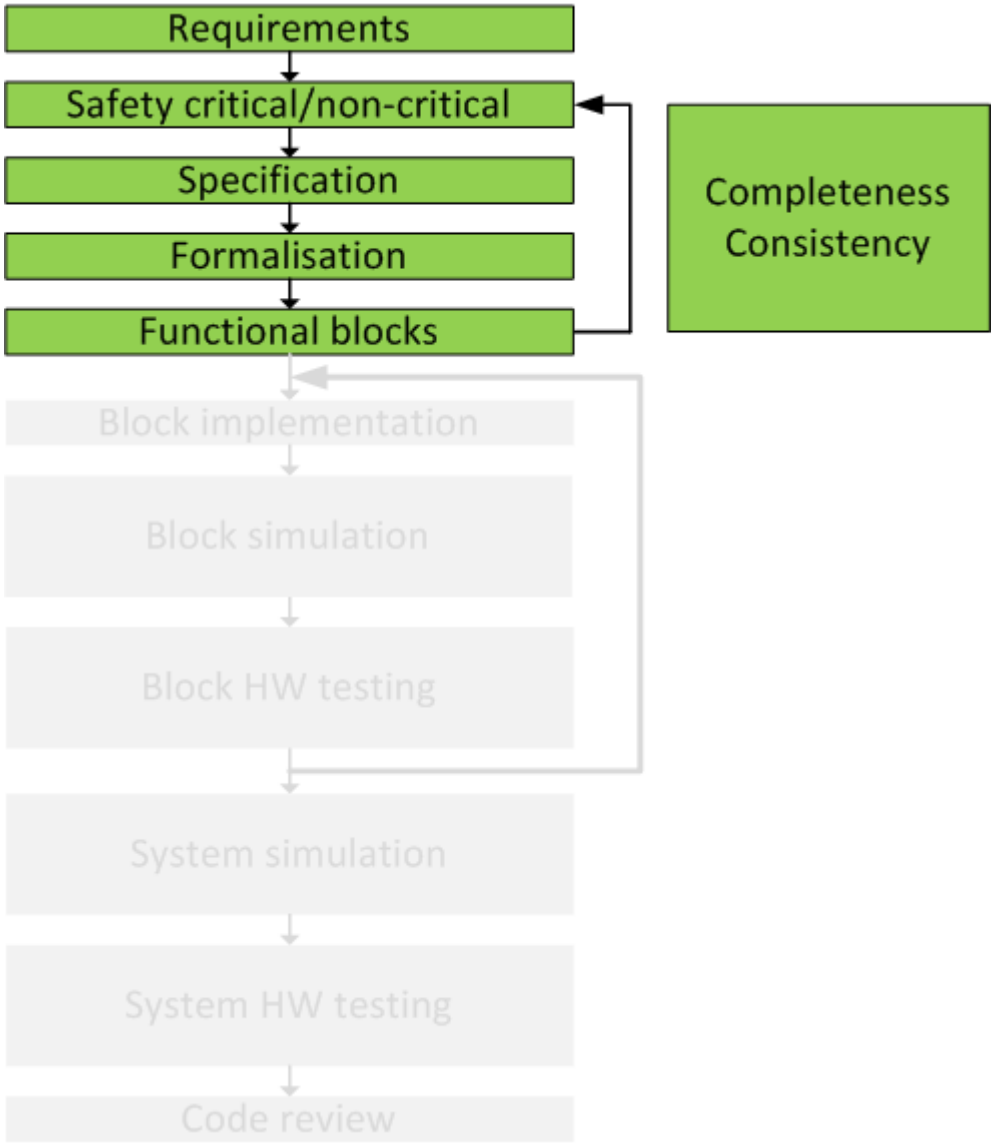
Arbiter – CISA

Control FPGA

VHDL implementation
Safety approach?

Hardware
Dependable Design

Design flow


Requirements
Safety critical/non-critical
Specification
Formalisation
Functional blocks
Completeness
Consistency
Block implementation
Block simulation
Block HW testing
System simulation
System HW testing
Code review

Requirements

Requirements requested by operators and/or approved by MPP.
E.G. Set-up Beam Flag equation

$$\left(\frac{E\,[GeV]}{450\,[GeV]}\right)^{1.7} \times I\,[p] \leq 1\times10^{12}$$ normal

$$\left(\frac{E\,[GeV]}{450\,[GeV]}\right)^{1.7} \times I\,[p] \leq 5\times10^{12}$$ relaxed

$$\left(\frac{E\,[GeV]}{450\,[GeV]}\right)^{1.7} \times I\,[p] \leq 1.3\times10^{13}$$ very relaxed

$$\left(\frac{E\,[GeV]}{450\,[GeV]}\right)^{1.7} \times I\,[p] \leq 2.5\times10^{11}$$ ion

Specification and formalisation

English language vs formal language

English + diagrams

2.3 PARAMETER CALCULATION, THRESHOLDS AND TRIMS

2.3.1 SPS_BCT4_INTENSITY CALCULATION

SPS_BCT4_INTENSITY is derived from SPS_BCT4_INTENSITY_A and _B. This implementation is to be made using a high-availability approach, **at least one** of the two intensity sources must be functional, SPS_BCT4_INTENSITY_A is to be used when it is VALID[1]. If it is NOT VALID, then the value of SPS_BCT4_INTENSITY_B is to be used. Finally, if neither intensity is VALID, the fail safe value is to be applied.

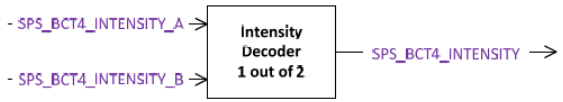


Figure 6 : SPS_BCT4_INTENSITY derived from _A and _B Values.

2.3.2 SPS PROBE BEAM FLAG CALCULATION

Parameter	SPS_PBF
Data Type [Default Value]:	Boolean [FALSE]
Source Data:	SPS_BCT4_INTENSITY
Related Threshold [Default Value]:	PROBE_BEAM_LIMIT [1 x 10^{11} p]
Related Trim:	OPERATOR_PROBE_BEAM_LIMIT

Calculation: SPS_PBF = TRUE when (SPS_BCT4_INTENSITY ≤ PROBE_BEAM_LIMIT) else SPS_PBF = FALSE

PROBE_BEAM_LIMIT may be trimmed by setting OPERATOR_PROBE_BEAM_LIMIT, the lower value of PROBE_BEAM_LIMIT and OPERATOR_PROBE_BEAM_LIMIT will be used in the calculation of the SPS_PBF.

predicate logic

```
FAIL_SAFE_LOW_INTENSITY: int := 1.6777215 * 10^15;

SPS_SMPC_PBF (OPERATOR_PROBE_BEAM_LIMIT : int,
          OPERATOR_PROBE_BEAM_LIMIT_is_set : bool,
          PROBE_BEAM_LIMIT : int,
          SPS_BCT4_INTENSITY_A : int,
          SPS_BCT4_INTENSITY_A_is_valid : bool,
          SPS_BCT4_INTENSITY_B : int,
          SPS_BCT4_INTENSITY_B_is_valid : bool,
          SPS_PBF : bool
         ) :=

        exists SPS_BCT4_INTENSITY : int. (
           (SPS_BCT4_INTENSITY =
            if SPS_BCT4_INTENSITY_A_is_valid then SPS_BCT4_INTENSITY_A;
            else if SPS_BCT4_INTENSITY_B_is_valid then SPS_BCT4_INTENSITY_B;
            else FAIL_SAFE_LOW_INTENSITY;
           )
         AND

           /* if operator probe beam limit is set, use it to trim probe beam limit */
        exists LIMIT : int. (
           (LIMIT =
             if NOT(OPERATOR_PROBE_BEAM_LIMIT_is_set) then PROBE_BEAM_LIMIT;
             else min(PROBE_BEAM_LIMIT, OPERATOR_PROBE_BEAM_LIMIT);
           )
         AND

           SPS_PBF =
             if SPS_BCT4_INTENSITY <= LIMIT then TRUE;
             else FALSE
        ));
```

Unlike the English, there is only one way to understand formal language.

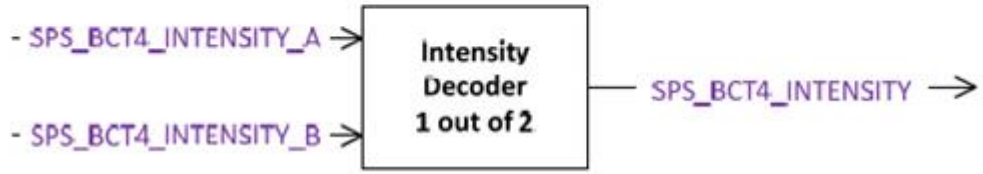
Specification and formalisation

English language vs formal language

English + diagrams

predicate logic

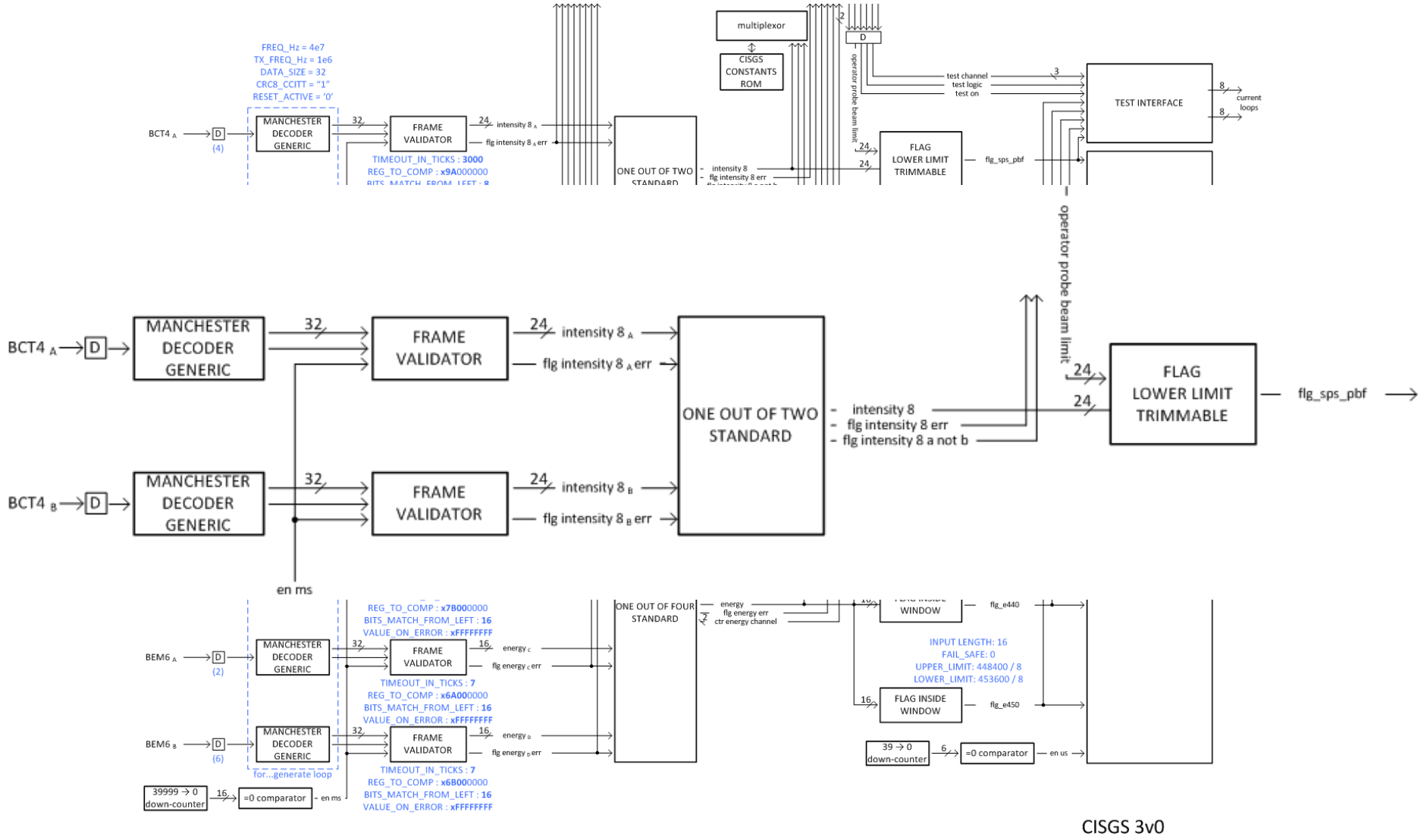
SPS_BCT4_INTENSITY is derived from SPS_BCT4_INTENSITY_A and _B. This implementation is to be made using a high-availability approach, **at least one** of the two intensity sources must be functional, SPS_BCT4_INTENSITY_A is to be used when it is VALID[1]. If it is NOT VALID, then the value of SPS_BCT4_INTENSITY_B is to be used. Finally, if neither intensity is VALID, the fail safe value is to be applied.



Data Type [Default Value]: Boolean [FALSE]
Source Data: SPS_BCT4_INTENSITY
Related Threshold [Default Value]: PROBE_BEAM_LIMIT [1 x 10^{11} p]
Related Trim: OPERATOR_PROBE_BEAM_LIMIT

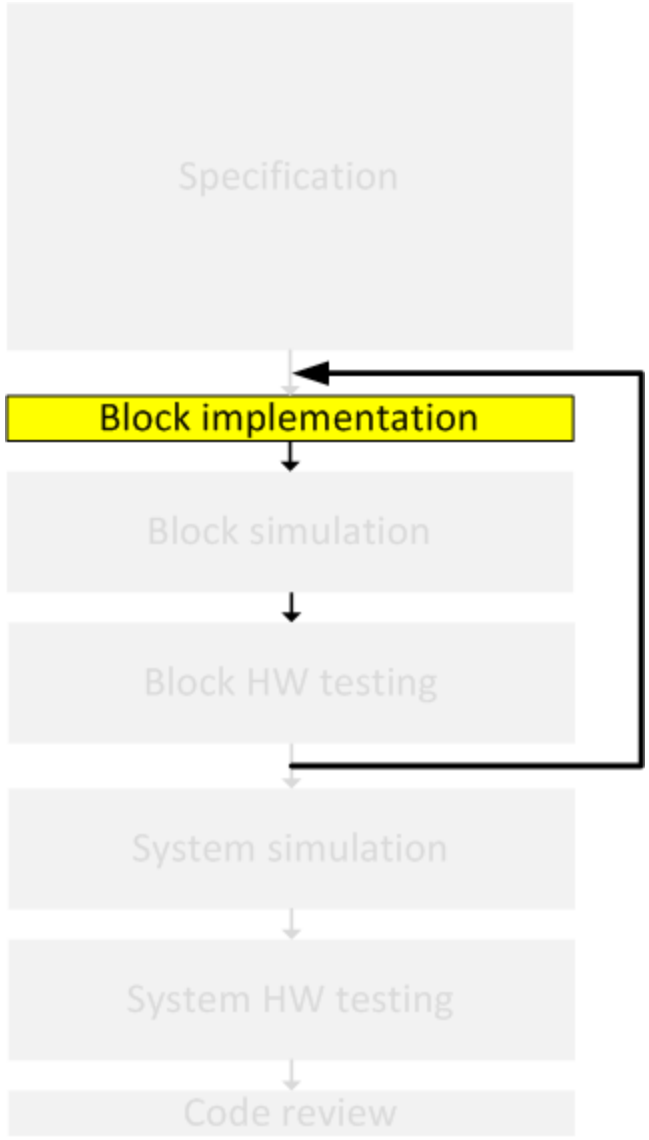
```
exists SPS_BCT4_INTENSITY : int. (
   (SPS_BCT4_INTENSITY =
    if SPS_BCT4_INTENSITY_A_is_valid then SPS_BCT4_INTENSITY_A;
    else if SPS_BCT4_INTENSITY_B_is_valid then SPS_BCT4_INTENSITY_B;
    else FAIL_SAFE_LOW_INTENSITY;
   )
 AND
```

Functional blocks



CISGS 3v0

Design flow


Specification
Block implementation
Block simulation
Block HW testing
System simulation
System HW testing
Code review

Implementation

```
--------------------------------------------------------------------------------
-- code and transmit data to the CISA
-- transmit 9 frames every 2 ms
--------------------------------------------------------------------------------

-- CISA frames rate interval counter (125 us)
us125_cnt_pr : process(rst, clk)
begin
    if rst = RESET_ACTIVE then
        us125_cnt <= us125_cnt_max;
        us125_cnt_en_d1 <= '0';
    elsif rising_edge (clk) then
        if us125_cnt_en = '0' then
            us125_cnt <= us125_cnt - 1;
        else
            us125_cnt <= us125_cnt_max;
        end if;
        us125_cnt_en_d1 <= us125_cnt_en;
    end if;
end process;
us125_cnt_en <= '1' when us125_cnt = 0 else '0';

-- CISA frames counter
cisa_frames_cnt_pr : process(rst, clk)
begin
    if rst = RESET_ACTIVE then
        cisa_frames_cnt <= "0000";
    elsif rising_edge (clk) then
        -- increase frame number after me is started
        if us125_cnt_en_d1 = '1' then
            if cisa_frames_cnt = "1111" then
                cisa_frames_cnt <= "0000";
            else
                cisa_frames_cnt <= cisa_frames_cnt + 1;
            end if;
        end if;
    end if;
end process;

-- start ME only when there is valid frame to transmit for selected time slot
cisa_meir <= us125_cnt_en and cisa_mei(32);
```

VHDL is not a programming language.
It is a Hardware Description Language

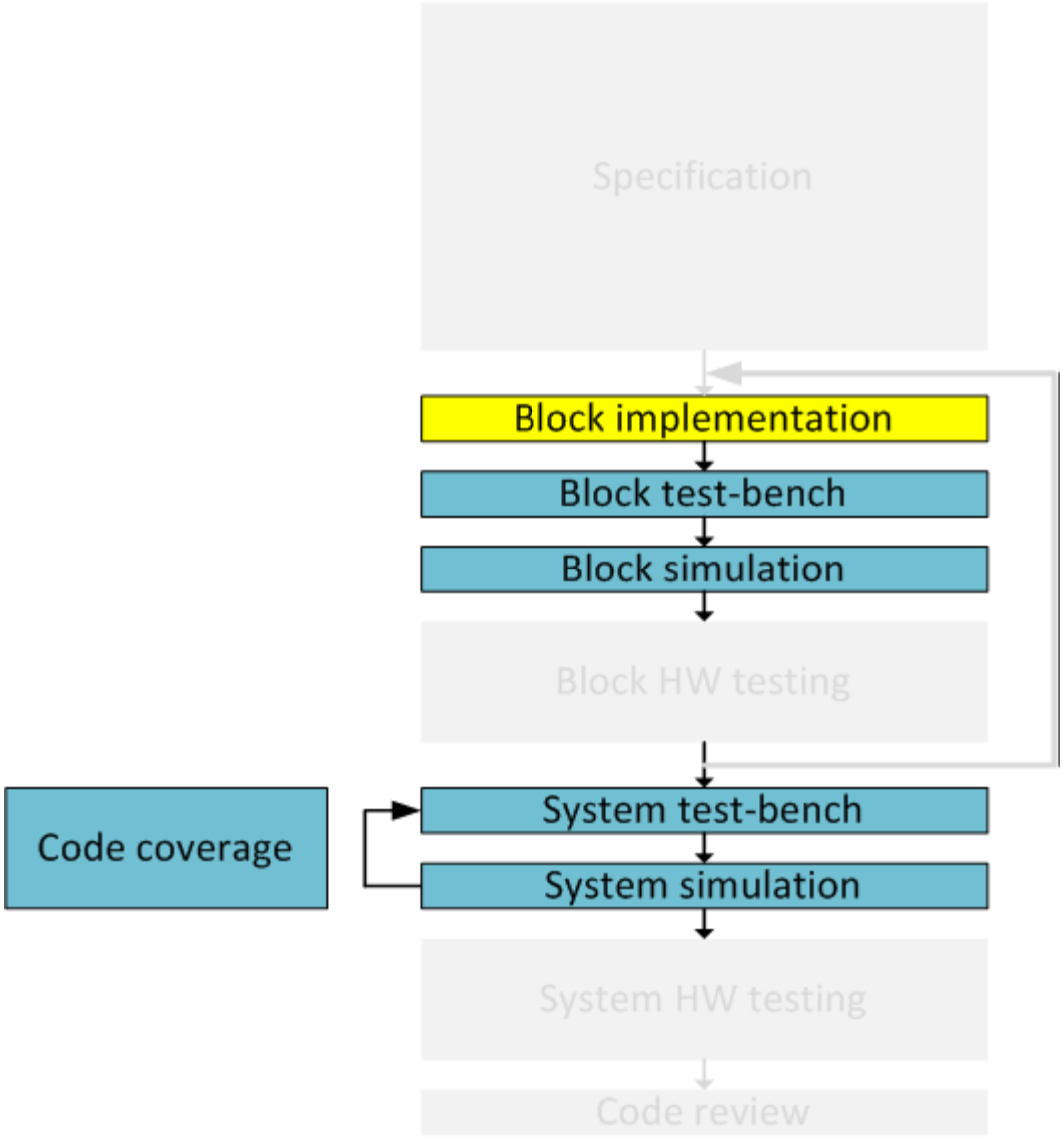
Must understand expected synthesis result

comments and naming convention
important for the code review

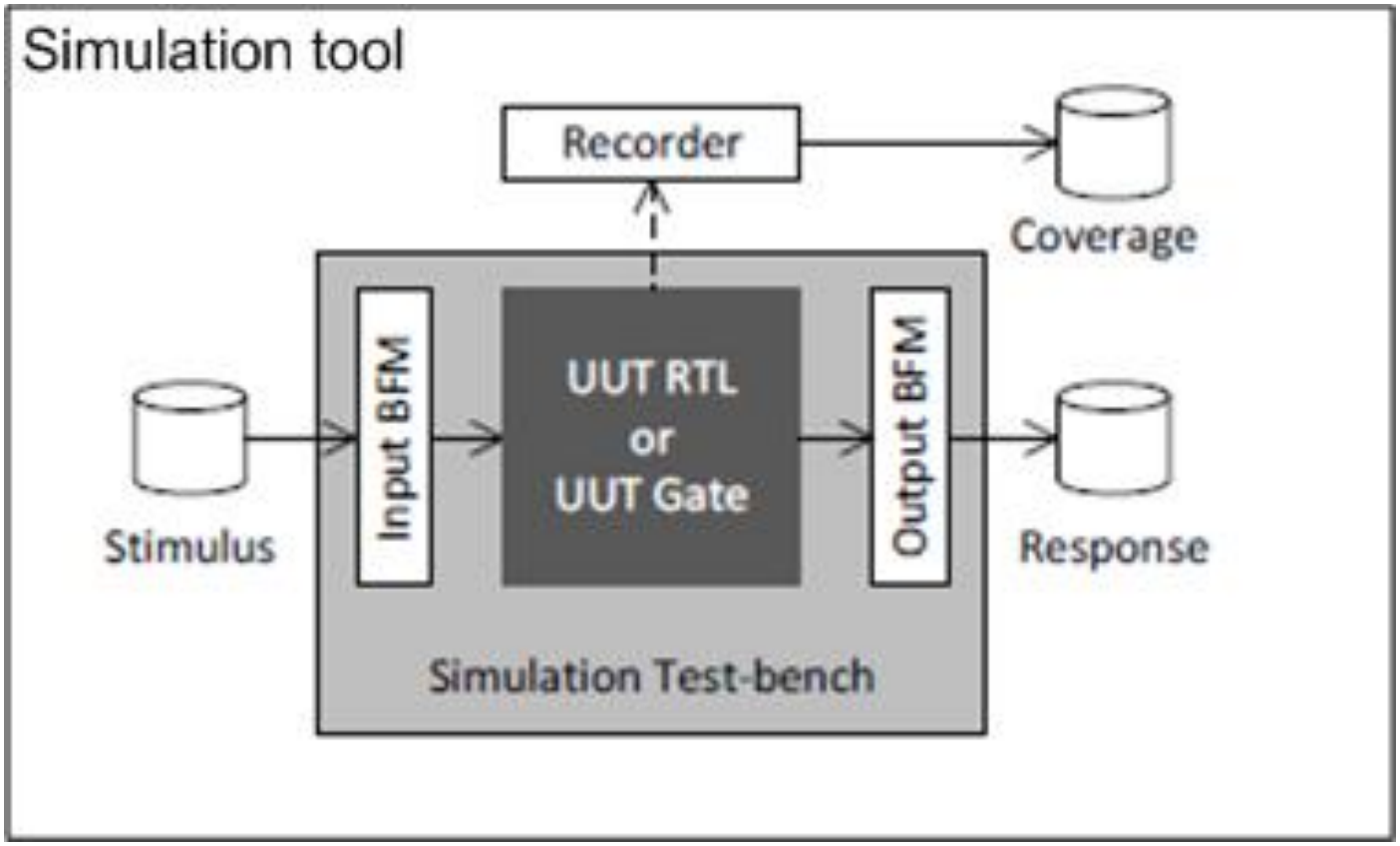
Critical code = strict
Non-Critical code = engineer has freedom

High % code reuse

Design flow

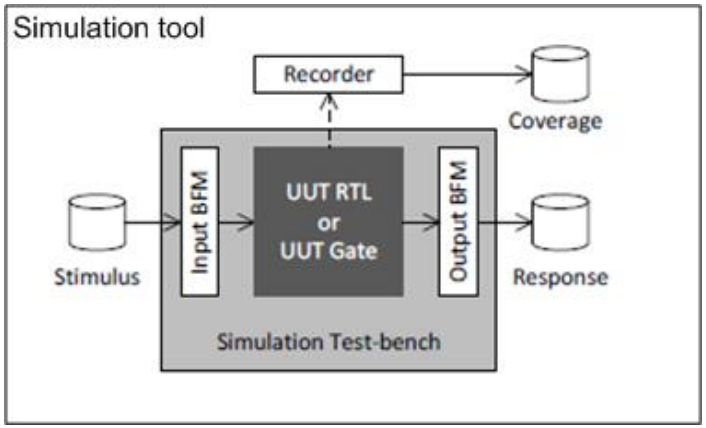

Specification
Block implementation
Block test-bench
Block simulation
Block HW testing
Code coverage
System test-bench
System simulation
System HW testing
Code review

Simulation



Unit Under Test
Bus Functional Model
Register Transfer Level

Simulation



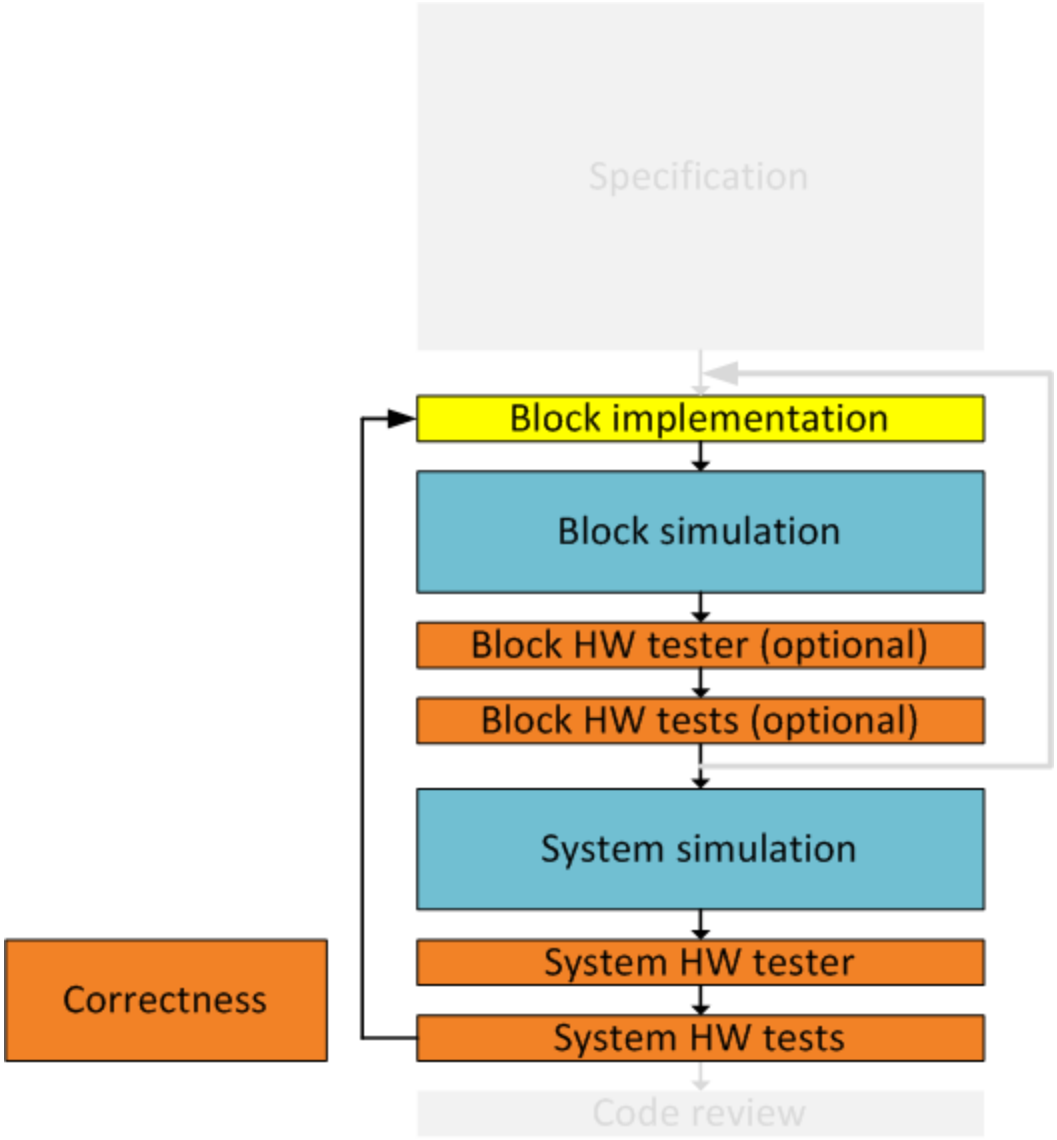
Test-bench = software wrapped around model

response should be correct for all stimulus

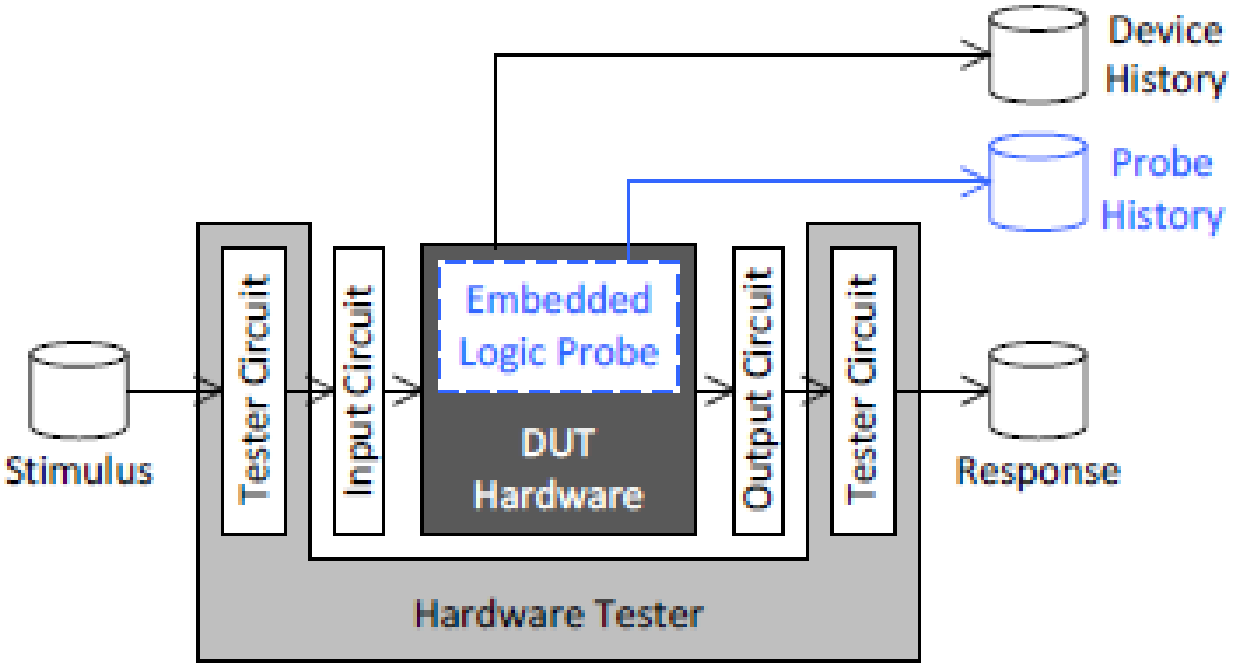
Simulation tool can examine code coverage

```
410  ------------------------------------------------------------------
411  -- stop error detection and register
412  ------------------------------------------------------------------
413  s_expected_p: process (clk, rst) begin
414    if rst = RESET_ACTIVE then
415      stop_expected <= '0';
416    elsif rising_edge(clk) then
417      --reset error when start condition is detected
418      if start_en = '1' and started_flg = '0' then
419        stop_expected <= '0';
420      --set error when all frame is received and stop condition wasn't detected
421      elsif started_flg = '1' and bits_cnt_done = '1' and sdata_en = '1' and bitv = '0' then
422        stop_expected <= '1';
423      end if;
424    end if;
425  end process;
426
```

Design flow

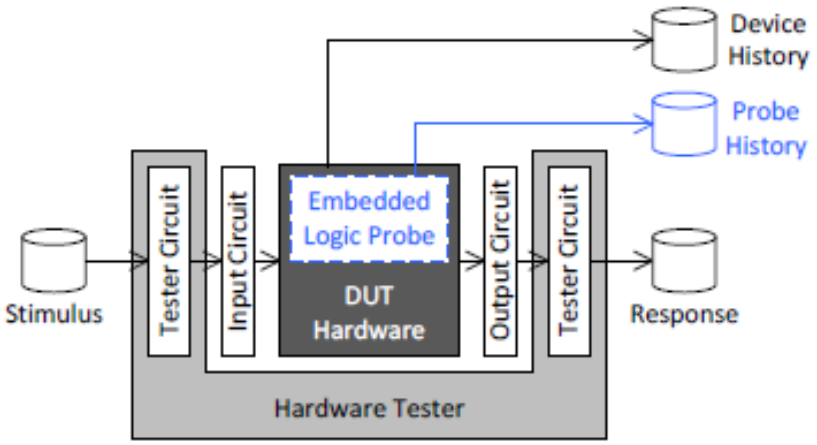

Specification
Block implementation
Block simulation
Block HW tester (optional)
Block HW tests (optional)
System simulation
System HW tester
System HW tests
Code review
Correctness

Hardware tester



DeviceUnder Test

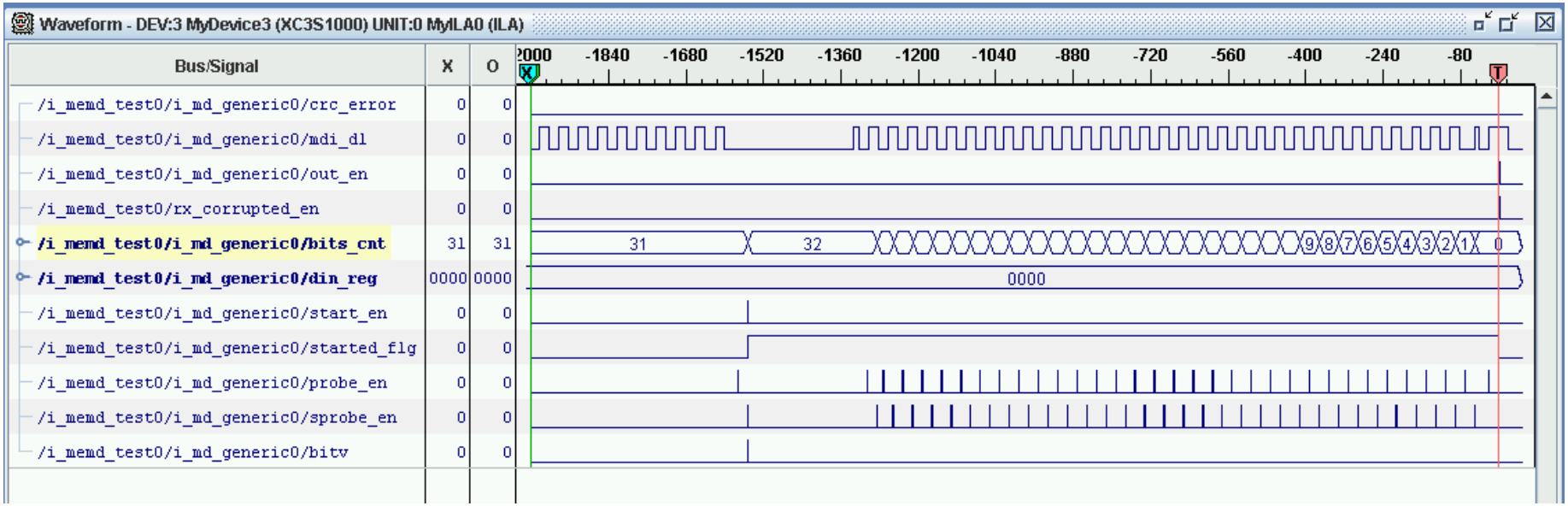
Hardware tester



similar to simulation but real hardware

Hardware response should be correct for each stimulus

embedded logic analyzers provided by FPGA vendors
Chip Scope, SignalTap, …



Hardware tester vs. simulation

Complementary

Software simulation:

source code tracking

code coverage

Hardware tester:

real time

real distortions

Hardware Dependable Design Summary

Our approach – dependable PLD design goes on top of dependable electronics design

formalisation of the specification

split critical – non-critical

reduction to minimum function

exhaustive source code simulation

full code coverage

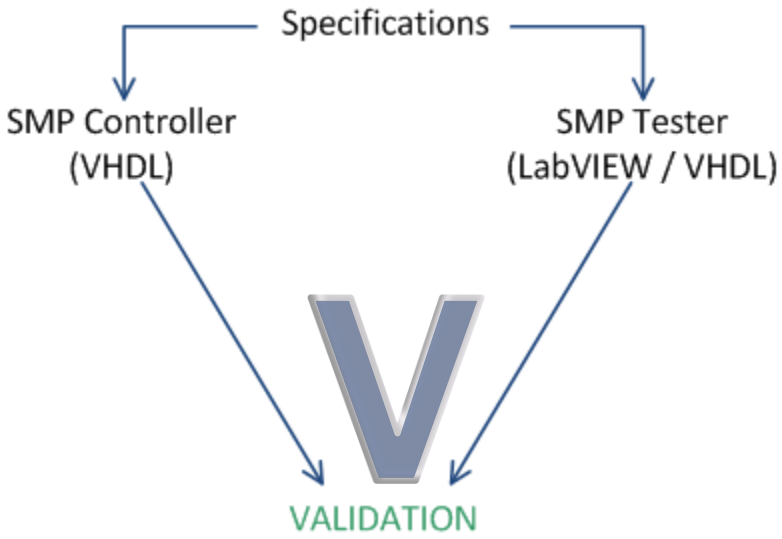
hardware testers

code reviews

external reviews

System Testing & Testers
The “V” Approach

SMP Development: the “V” approach



English Specification used for the Tester

Determine Tests needed to verify each function

Developed Independently of Controller

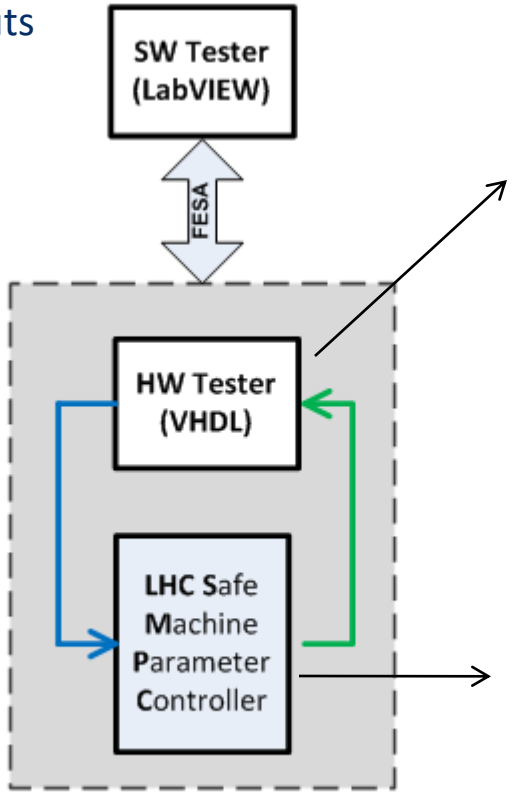
Validation of Controller versus Tester versus English Specification

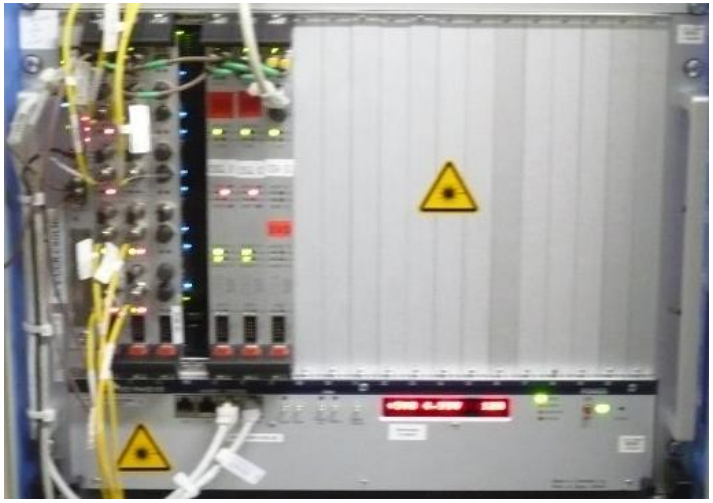
What is the SMP Tester?

Definition : Ensure the SMP controller works as specified

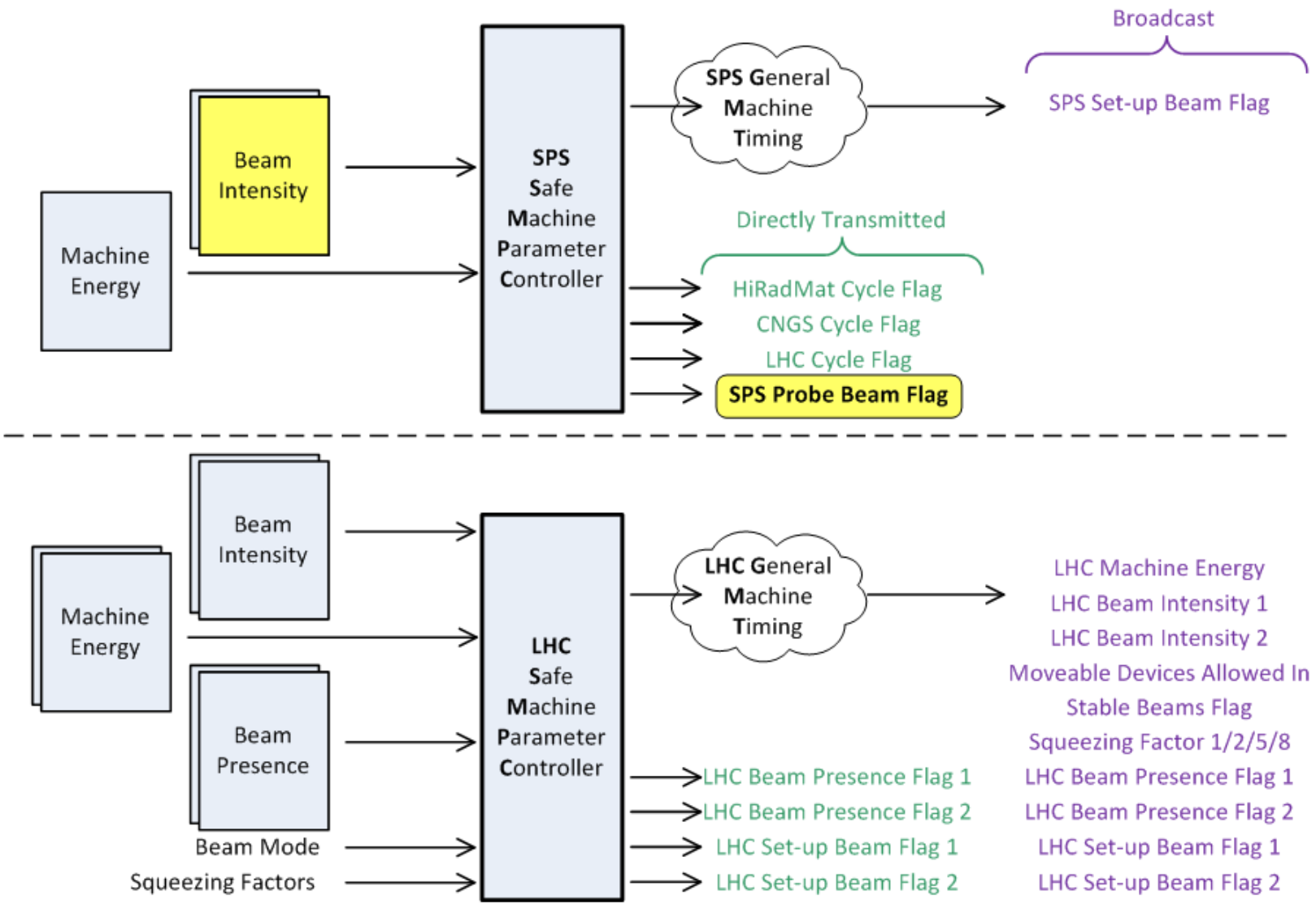
Roles :

- Simulates the inputs
- Analyzes the outputs

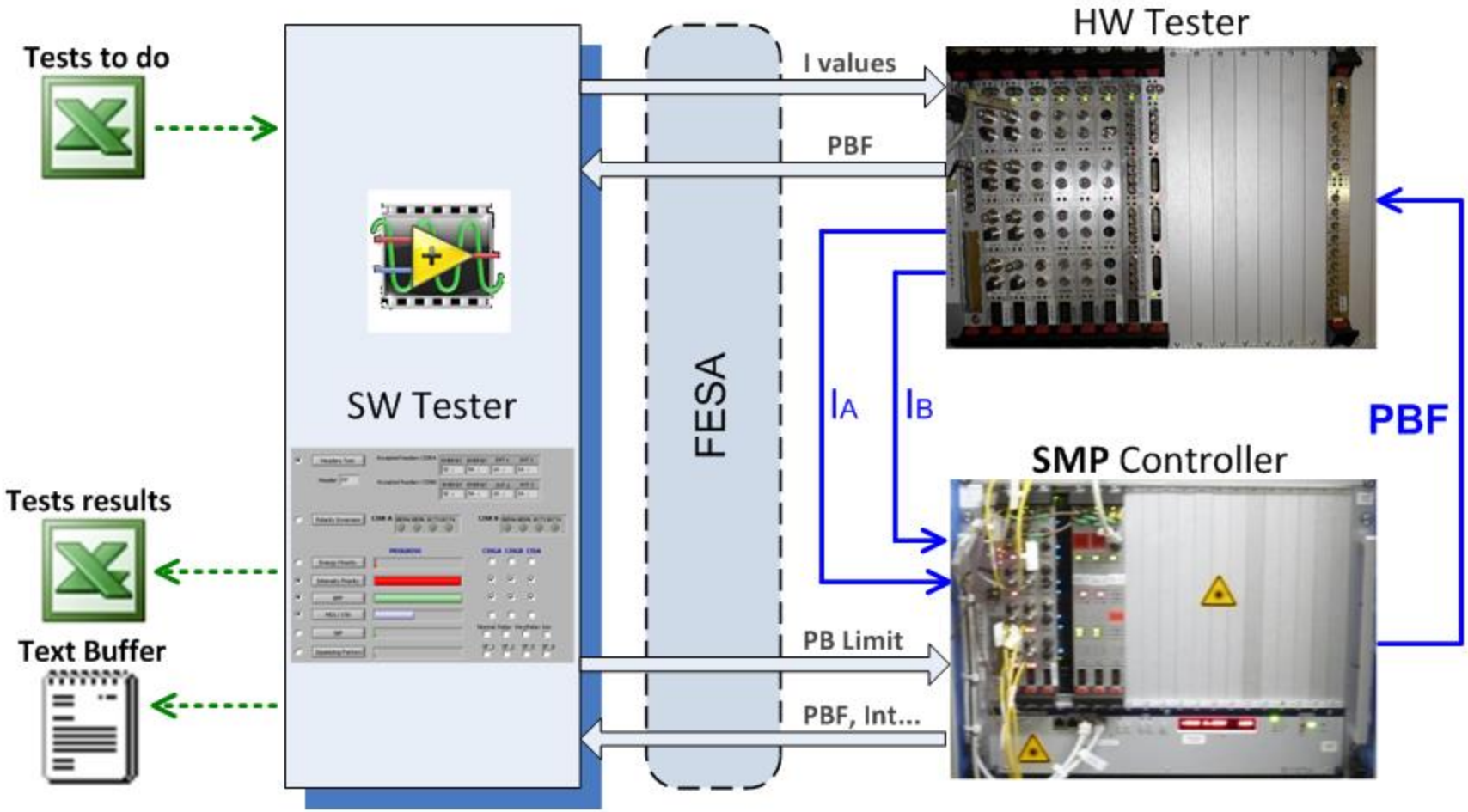




Functionalities of the SMP tester

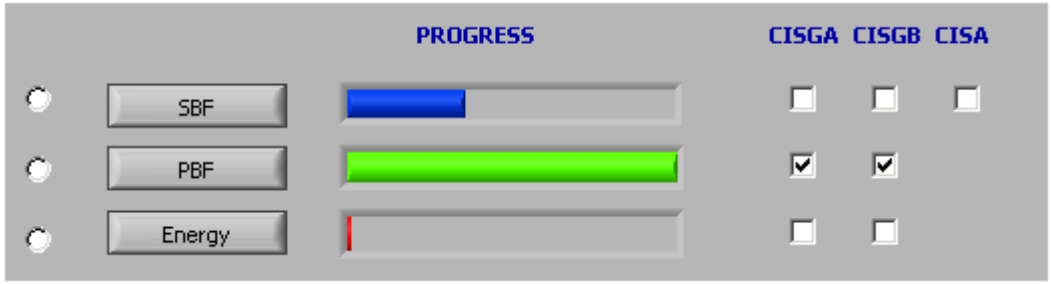


SPS Probe-Beam Flag: Test Protocol



SPS Probe-Beam Flag: Display Results

LabVIEW SubPanel

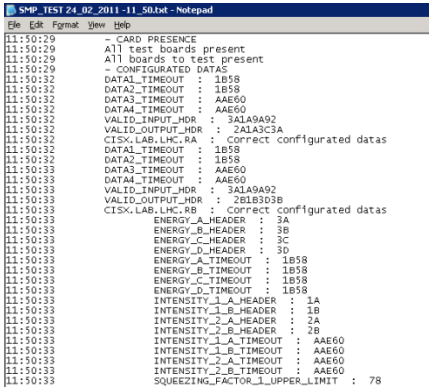


Excel file

RESULTS GA

Last Data Received A	Last Data Received B	I10_A Error/NoData	I10_B Error/NoData	Intensity	PBF	Value after 3 sec	HB Timing	Correct
8A007772	8B007772			7772	0	0	0	T
8A007772	8B007772			FFFF	0	0	0	T
8A007772	8B007772			FFFF	0	0	0	T

Text File



```
SMP_TEST 24_02_2011 -11_50.txt - Notepad
File  Edit  Format  View  Help
11:50:29        - CARD PRESENCE
11:50:29        All test boards present
11:50:29        All boards to test present
11:50:29        - CONFIGURATED DATAS
11:50:32        DATA1_TIMEOUT  :  1B58
11:50:32        DATA2_TIMEOUT  :  1B58
11:50:32        DATA3_TIMEOUT  :  AAE60
11:50:32        DATA4_TIMEOUT  :  AAE60
11:50:32        VALID_INPUT_HDR  :  3A1A9A92
11:50:32        VALID_OUTPUT_HDR  :  2A1A3C3A
11:50:32        CISX.LAB.LHC.RA  :  Correct configurated datas
11:50:32        DATA1_TIMEOUT  :  1B58
11:50:32        DATA2_TIMEOUT  :  1B58
11:50:33        DATA3_TIMEOUT  :  AAE60
11:50:33        DATA4_TIMEOUT  :  AAE60
11:50:33        VALID_INPUT_HDR  :  3A1A9A92
11:50:33        VALID_OUTPUT_HDR  :  2B1B3D3B
11:50:33        CISX.LAB.LHC.RB  :  Correct configurated datas
11:50:33                ENERGY_A_HEADER  :  3A
11:50:33                ENERGY_B_HEADER  :  3B
11:50:33                ENERGY_C_HEADER  :  3C
11:50:33                ENERGY_D_HEADER  :  3D
11:50:33                ENERGY_A_TIMEOUT  :  1B58
11:50:33                ENERGY_B_TIMEOUT  :  1B58
11:50:33                ENERGY_C_TIMEOUT  :  1B58
11:50:33                ENERGY_D_TIMEOUT  :  1B58
11:50:33                INTENSITY_1_A_HEADER  :  1A
11:50:33                INTENSITY_1_B_HEADER  :  1B
11:50:33                INTENSITY_2_A_HEADER  :  2A
11:50:33                INTENSITY_2_B_HEADER  :  2B
11:50:33                INTENSITY_1_A_TIMEOUT  :  AAE60
11:50:33                INTENSITY_1_B_TIMEOUT  :  AAE60
11:50:33                INTENSITY_2_A_TIMEOUT  :  AAE60
11:50:33                INTENSITY_2_B_TIMEOUT  :  AAE60
11:50:33                SQUEEZING_FACTOR_1_UPPER_LIMIT  :  78
```

Tester Summary

What it does:

- Replaces all elements connected to the SMP

- Tests automatically many input combinations

- Validates the boards for the operation

Software
FESA – RBAC – MCS – Checks - GUI

Introduction

FESA class

RBAC protection and MCS

Operational checks

SMP-GUI

FESA class

FESA class provides access to hardware registers

no complex logic behind, just valid range checks

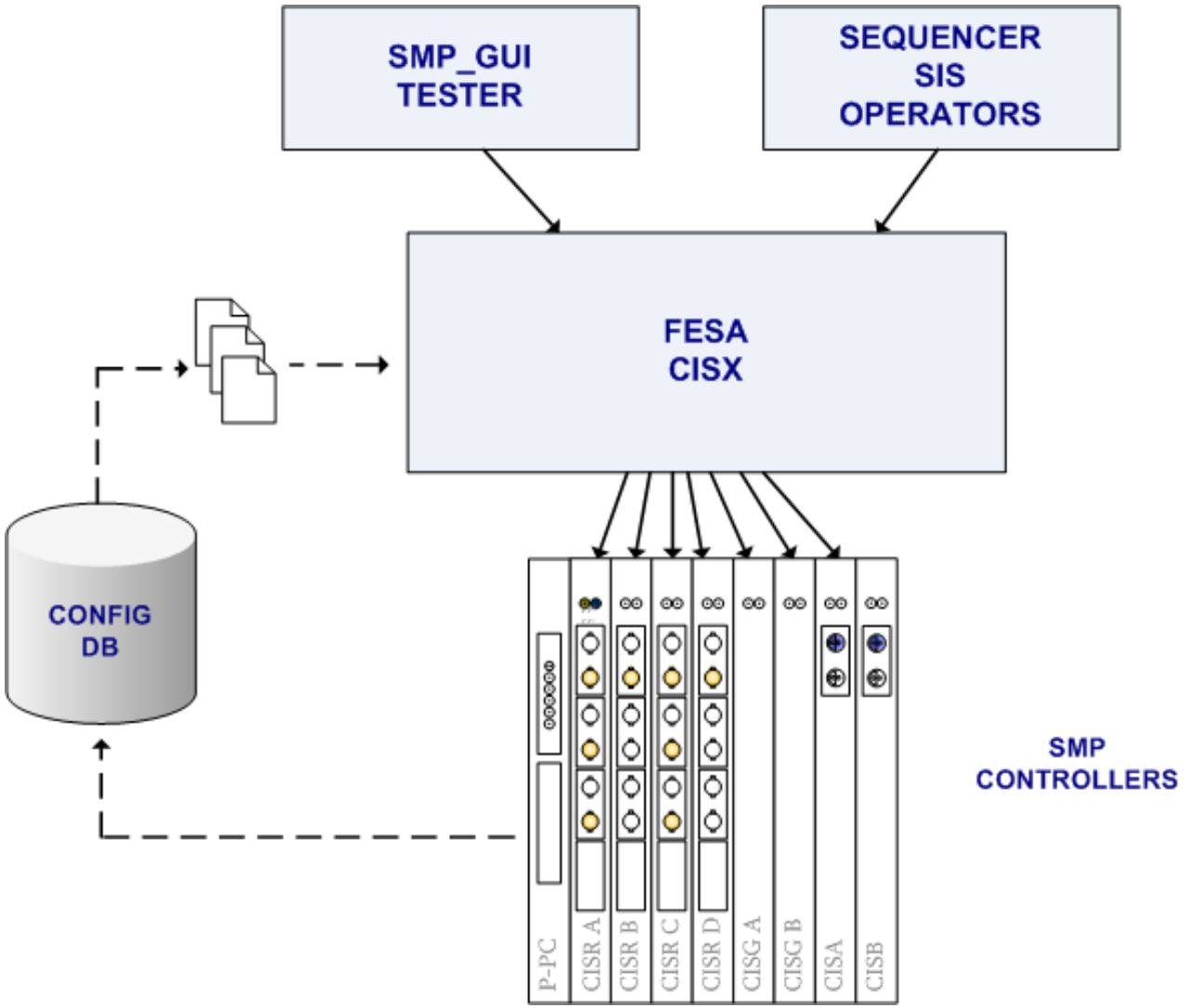
Different type of access

read-only access for everyone

write access for experts trough dedicated expert properties

write access for critical registers for operation

FESA class


SMP_GUI
TESTER
SEQUENCER
SIS
OPERATORS
FESA
CISX
CONFIG
DB
P-PC
CISR A
CISR B
CISR C
CISR D
CISG A
CISG B
CISA
CISB
SMP
CONTROLLERS

RBAC and MCS configuration

SPS

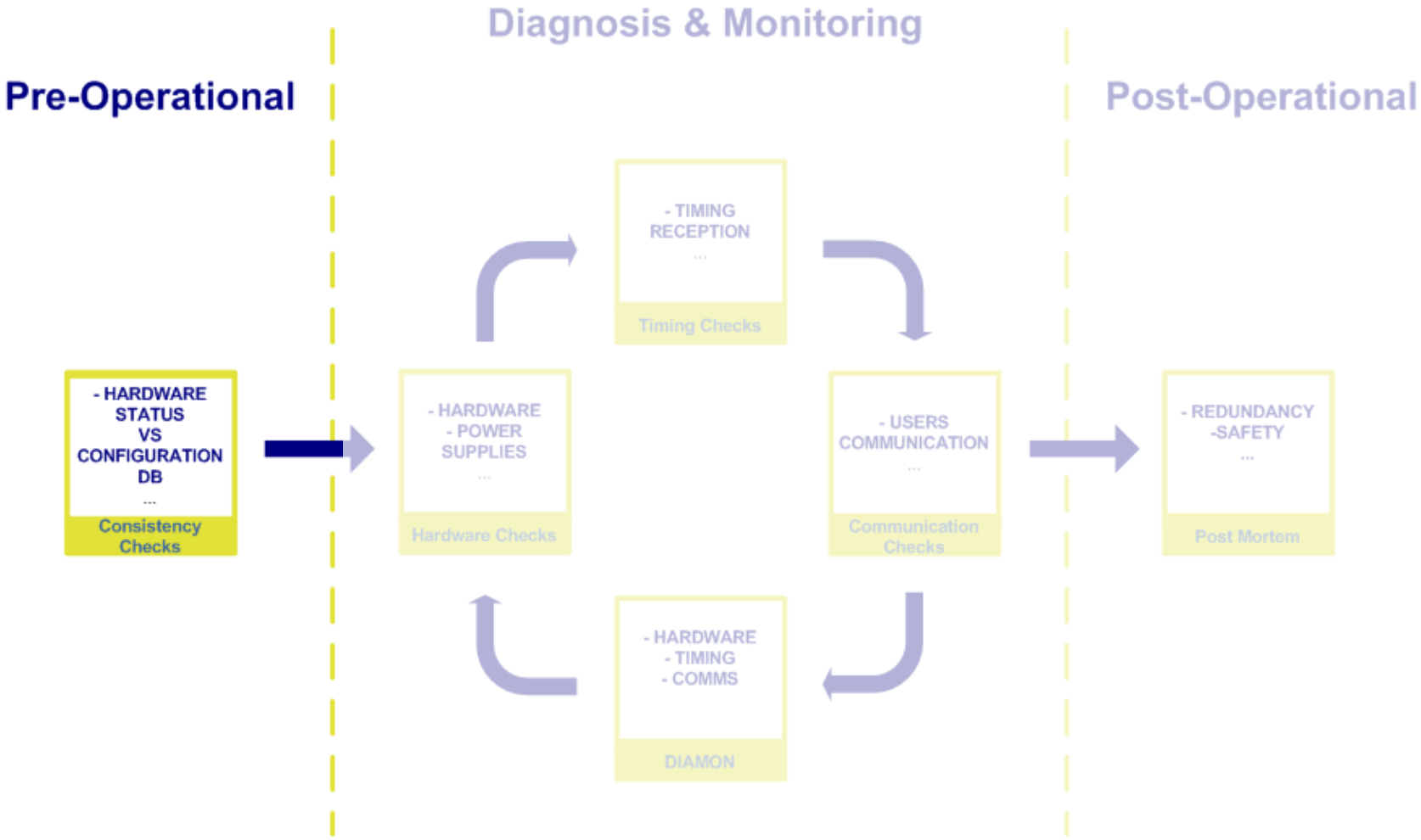
Property	Roles	Applications	Location	MCS
ProbeBeamLimit	LHC-OP, LHC-EIC, MCS-SMP	SEQUENCER		X

LHC

Property	Roles	Applications	Location	MCS
SqueezingFactor	LHC-OP, LHC-EIC, MCS-SMP	SIS	SIS-HOSTS	
SqueezingFactorLimits	LHC-OP, LHC-EIC, MCS-SMP	SEQUENCER		X
PhysicsEnergyLimits	LHC-OP, LHC-EIC, MCS-SMP	SEQUENCER		X
BeamMode	LHC-OP, LHC-EIC, MCS-SMP	SEQUENCER		
ForceSetupBeamFlag	LHC-OP, LHC-EIC, MCS-SMP	SMP-GUI		
SetupBeamFlagNormal	LHC-OP, LHC-EIC, MCS-SMP	SMP-GUI		
SetupBeamFlagSpecial	SMP-THRESHOLD-EXPERT	SMP-GUI		
ExpertRegisterSetting	SMP-EXPERT	SMP-GUI		

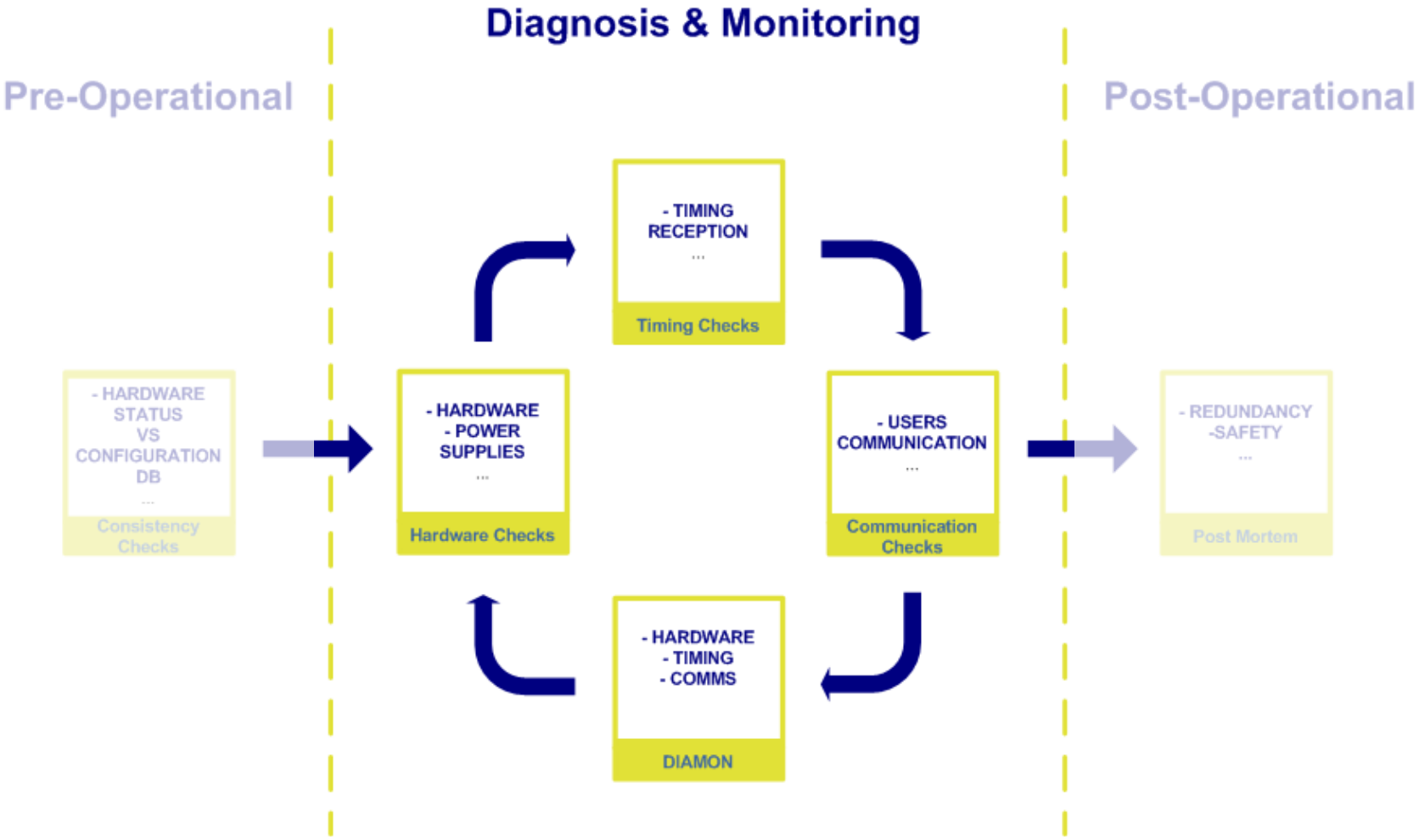
Operational checks

Pre-operational checks to ensure system ready for operation
HW consistency vs DB, Test mode to ensure critical paths working to spec…



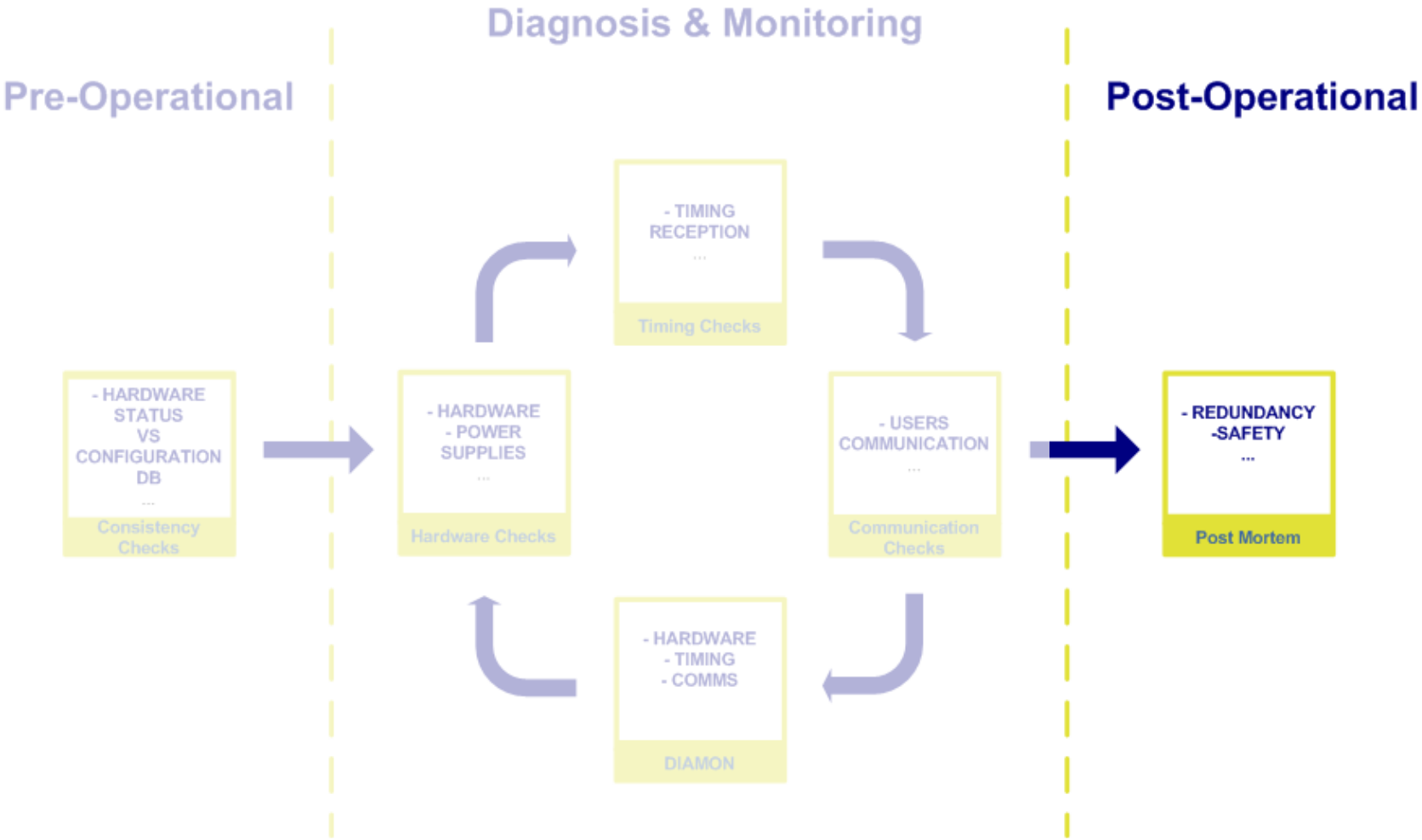
Operational checks

DIAMON checks to detect infrastructure issues
PS, Timing, Communication problems…



Operational checks

Post-Mortem for post-operational check sequence
Role played in last dump, Redundancy, Safety for next mission…



GUI
Demonstration

SMP-GUI

GUI to monitor status of the systems (SPS and LHC)

Send commands to the controllers

Logged data viewer

Useful tool for diagnostics

Same tool used for Operators and Experts:

Provide different functionalities depending on user roles

Status &
Future Plans

Q1/2 2011

+ ongoing documentation

+ study intensity logic

+ 10 trivial issues in monitoring and diagnostics

+ beta Pre-Op

+ beta DIAMON

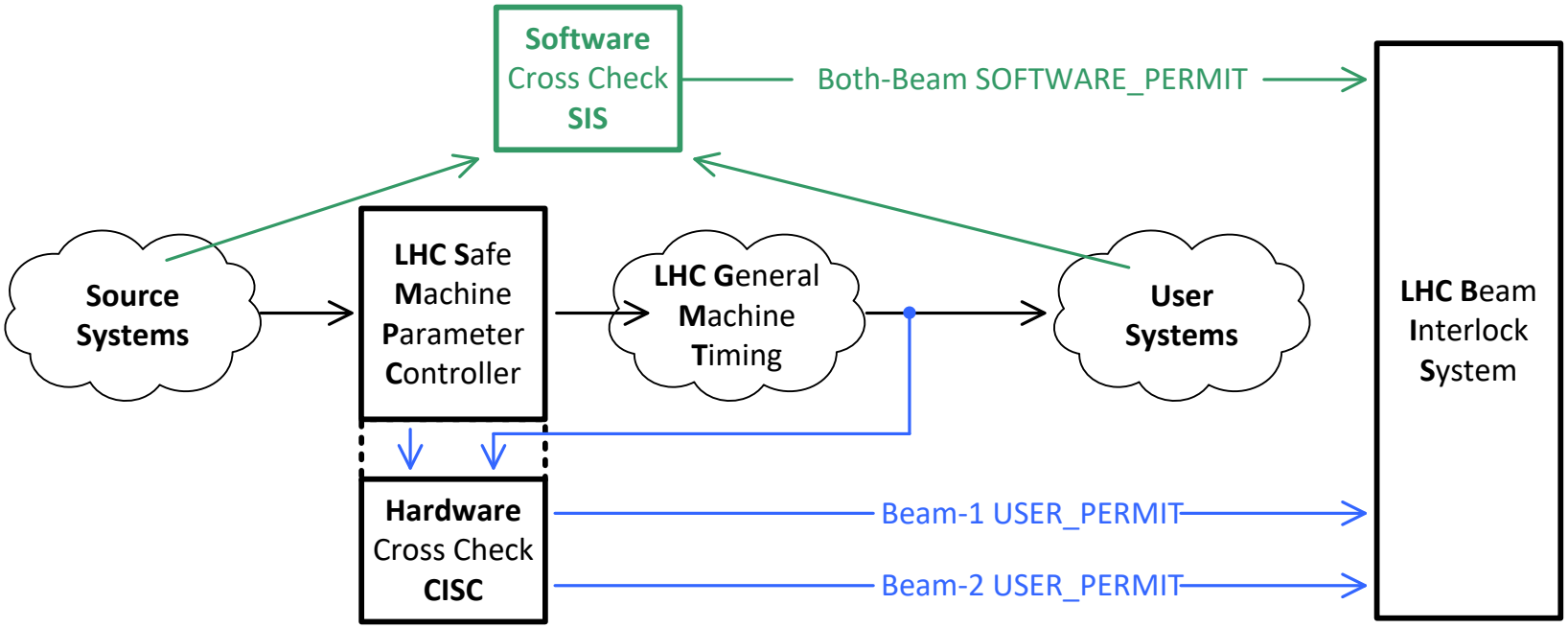
+ beta Post-Mortem

Q3/4 2011+

+ Cross-checking hardware

+ Cross-checker tester

Q3/4 2011+


Software
Cross Check
SIS
Both-Beam SOFTWARE_PERMIT
Source Systems
LHC Safe Machine Parameter Controller
LHC General Machine Timing
User Systems
LHC Beam Interlock System
Hardware
Cross Check
CISC
Beam-1 USER_PERMIT
Beam-2 USER_PERMIT

Q3/4 2011+

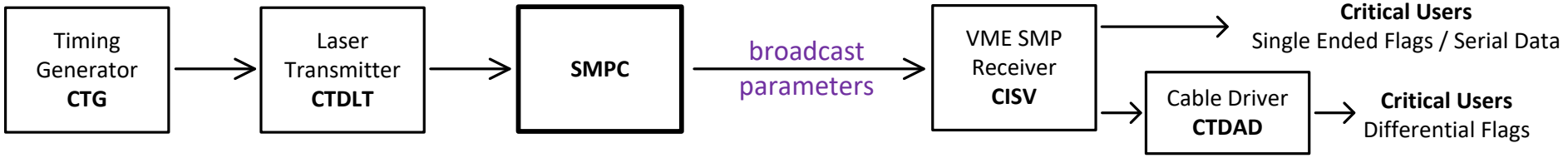
+ Cross-checking hardware

+ Cross-checker tester

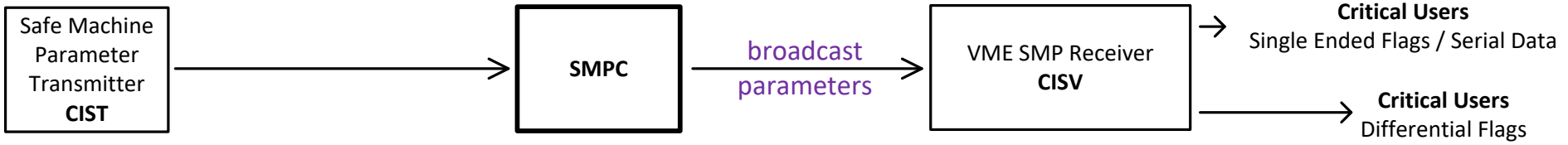
+ VME Receiver

+ VME Transmitter

Q3/4 2011+


Timing Generator **CTG**
Laser Transmitter **CTDLT**
SMPC
broadcast parameters
VME SMP Receiver **CISV**
Critical Users
Single Ended Flags / Serial Data
Cable Driver **CTDAD**
Critical Users
Differential Flags

Q3/4 2011+


Safe Machine Parameter Transmitter
CIST
SMPC
broadcast parameters
VME SMP Receiver
CISV
Critical Users
Single Ended Flags / Serial Data
Critical Users
Differential Flags

Q3/4 2011+

+ Cross-checking hardware

+ Cross-checker tester

+ VME Receiver

+ VME Transmitter

+ Pre-Op

+ DIAMON

+ Post-Mortem

Thank you!