



P. Fernandez Carmona:: Center for Proton Therapy:: Paul Scherrer Institut

Significant acceleration of development by **automating quality assurance** of a medical particle accelerator safety system using a formal language driven test stand

The problem: Patients safety system QA

Sequential stimuli

```
SIGNAL_A <= NOK;
wait for t_PAUSE;
SIGNAL_B <= OK;
```

Assertions

```
assert SIGNAL_C = NOK report "SIGNAL_C not in state NOK at start of test" severity FAILURE;
```

Measurements

```
measure falling_edge(SIGNAL_1) to rising_edge(SIGNAL_2)
name "t_BTL";
```



PSI PAUL SCHERRER INSTITUT PROSCAN

3. QA Release Testing - Tests & Results

Additional to the test results, the test results are evaluated against the test results. The test results are evaluated against the test results. The test results are evaluated against the test results.

3.1 QA Test Results

Test ID	Test Name	Test Date	Test Result	Test Status
1	Unit Test 1	2023-01-01	Pass	Completed
2	Unit Test 2	2023-01-02	Fail	Failed
3	Unit Test 3	2023-01-03	Pass	Completed
4	Unit Test 4	2023-01-04	Pass	Completed
5	Unit Test 5	2023-01-05	Pass	Completed
6	Unit Test 6	2023-01-06	Pass	Completed
7	Unit Test 7	2023-01-07	Pass	Completed
8	Unit Test 8	2023-01-08	Pass	Completed
9	Unit Test 9	2023-01-09	Pass	Completed
10	Unit Test 10	2023-01-10	Pass	Completed

PSI PAUL SCHERRER INSTITUT PROSCAN

3.1 QA Test Results

3.1.1 Test Results

3.1.2 Test Results

3.1.3 Test Results

3.1.4 Test Results

3.1.5 Test Results

3.1.6 Test Results

3.1.7 Test Results

3.1.8 Test Results

3.1.9 Test Results

3.1.10 Test Results

3.1.11 Test Results

3.1.12 Test Results

3.1.13 Test Results

3.1.14 Test Results

3.1.15 Test Results

3.1.16 Test Results

3.1.17 Test Results

3.1.18 Test Results

3.1.19 Test Results

3.1.20 Test Results

3.1.21 Test Results

3.1.22 Test Results

3.1.23 Test Results

3.1.24 Test Results

3.1.25 Test Results

3.1.26 Test Results

3.1.27 Test Results

3.1.28 Test Results

3.1.29 Test Results

3.1.30 Test Results

3.1.31 Test Results

3.1.32 Test Results

3.1.33 Test Results

3.1.34 Test Results

3.1.35 Test Results

3.1.36 Test Results

3.1.37 Test Results

3.1.38 Test Results

3.1.39 Test Results

3.1.40 Test Results

3.1.41 Test Results

3.1.42 Test Results

3.1.43 Test Results

3.1.44 Test Results

3.1.45 Test Results

3.1.46 Test Results

3.1.47 Test Results

3.1.48 Test Results

3.1.49 Test Results

3.1.50 Test Results

3.1.51 Test Results

3.1.52 Test Results

3.1.53 Test Results

3.1.54 Test Results

3.1.55 Test Results

3.1.56 Test Results

3.1.57 Test Results

3.1.58 Test Results

3.1.59 Test Results

3.1.60 Test Results

3.1.61 Test Results

3.1.62 Test Results

3.1.63 Test Results

3.1.64 Test Results

3.1.65 Test Results

3.1.66 Test Results

3.1.67 Test Results

3.1.68 Test Results

3.1.69 Test Results

3.1.70 Test Results

3.1.71 Test Results

3.1.72 Test Results

3.1.73 Test Results

3.1.74 Test Results

3.1.75 Test Results

3.1.76 Test Results

3.1.77 Test Results

3.1.78 Test Results

3.1.79 Test Results

3.1.80 Test Results

3.1.81 Test Results

3.1.82 Test Results

3.1.83 Test Results

3.1.84 Test Results

3.1.85 Test Results

3.1.86 Test Results

3.1.87 Test Results

3.1.88 Test Results

3.1.89 Test Results

3.1.90 Test Results

3.1.91 Test Results

3.1.92 Test Results

3.1.93 Test Results

3.1.94 Test Results

3.1.95 Test Results

3.1.96 Test Results

3.1.97 Test Results

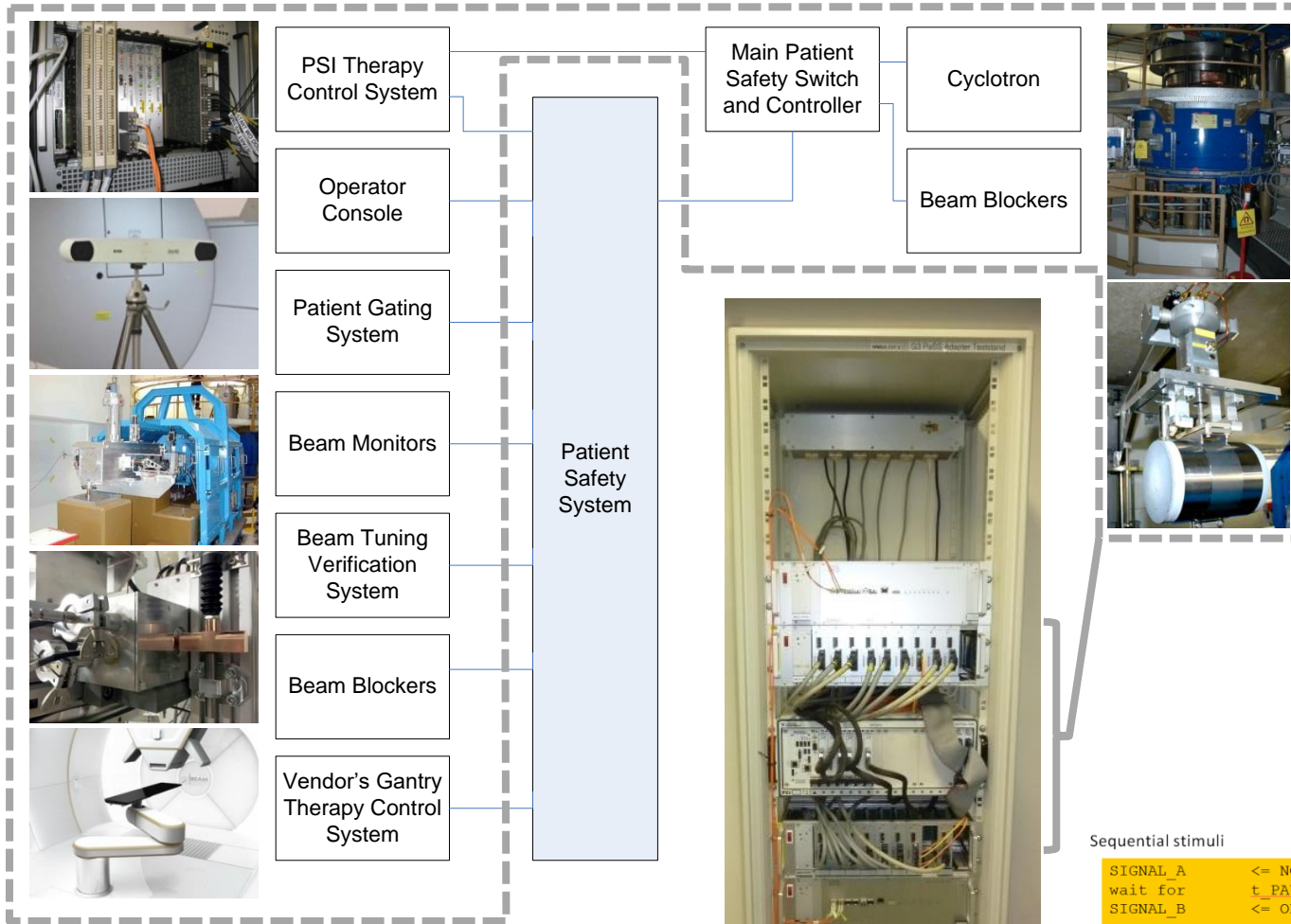
3.1.98 Test Results

3.1.99 Test Results

3.1.100 Test Results



The solution: Automated test bench for unit tests



Sequential stimuli

```
SIGNAL_A <= NOK;
wait for t_PAUSE;
SIGNAL_B <= OK;
```



Assertions

```
assert SIGNAL_C= NOK report " SIGNAL_C not in state NOK at
start of test" severity FAILURE;
```



Measurements

```
measure falling_edge(SIGNAL_1) to rising_edge(SIGNAL_2)
name "t_BTL";
```

