

Motivation & Challenger

- Develop an integrated framework that allows the reuse of verification test benches during all the design process steps.
- The source code of the test benches and the FUT (“Function Under Test”) are not modified during the verification process.
- The same source code is used for virtual platforms as well as on-board verification.

```

a) FUT source code
int imgProc( int nImg) { //FUT
for( n=0; n < nImg; n++) {
    fut1: read=readImg(img);
        if(read==0) break;
    fut2: processImg(img, Result);
    fut3: dispImg(Result);
}
return n;
}

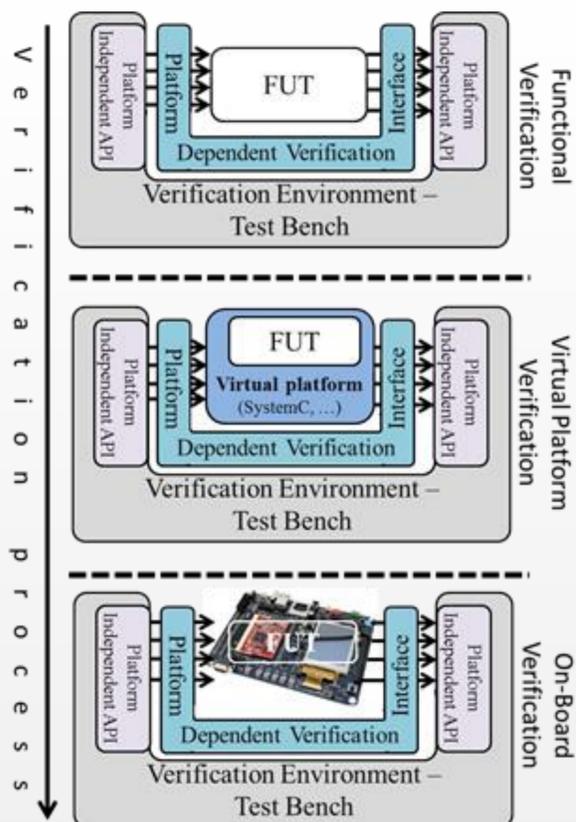
b) Test bench source code
numImages= 1; // Test File source code
TEST(test_case1, test1) { // test1 begin -> PDI
    tst1: EXPECT_CALL( readImg ).WillRepeatedly( RetLenalng );
    tst2: EXPECT_CALL( dispImg ).WillRepeatedly( CheckLena );
    tst3: int ret = imgProc( numImages ); //FUT execution
    tst4: EXPECT_EQ( ret, 1 );
} // test1 end

```

Main Contributions

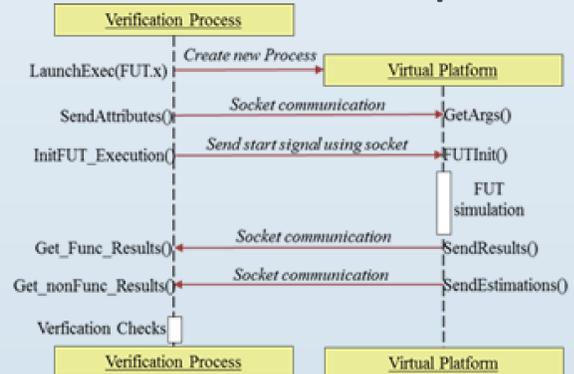
- Reusable verification environment integrates two independent test execution layers.
 - The Platform-Independent Layer (PIL) to be reused from one environment to another.
 - The Platform-Dependent Interface (PDI) to be executed in the verification platform.
- Communication protocol that enables information to be transferred between the PIL and PDI layers.
- Functionality of the PDI layer in different verification environments (virtual platform and on-board testing).

Proposed methodology

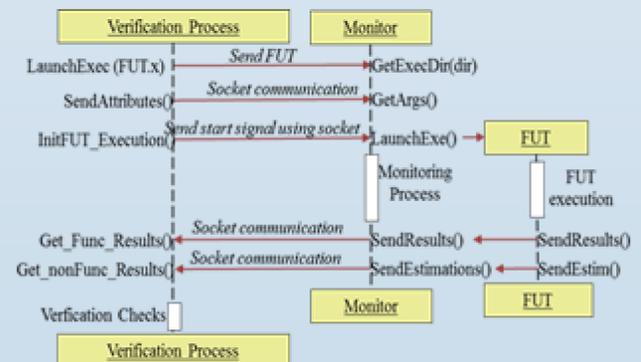


Implementation

Verification on virtual platform



Verification on board



Experimental results

Case Study 1: MediaBench programs

Verification level	Proposed methodology	Classical methodology
Functional	1060 lines	937 lines
Virtual Platform	3 modified lines	975 lines
On-board	3 modified lines	956 lines
Total	1076 lines	2868 lines

Case Study 2: Face recognizer

Test type	#test	#lines	Checked assertions
Unit	1.605	23.796	1.130.073 functional 3.210 non-functional
Integration	700	17.793	2.365.103 functional 1.400 non-functional
Acceptation	387	6.717	1.254 functional 752 non-functional
Accuracy	1.300	20.819	3.712 functional
TOTAL	3.992	69.125	3.500.142 functional 5.362 non-functional

Conclusions

- New methodology and framework to verify functional and non-functional requirements of embedded SW during all the stages of the design process.
- Enables the execution of tests on different verification environments such as virtual platforms or real HW boards.
- Enables the execution of tests on different verification levels: functional, virtual platform and on-board verification levels.
- Avoids rewriting the tests and reduce development effort and time-to-market.