Test Time Reduction for ECS testing with Dynamic DC/SmartCalc Method

Presented By: Stephen Fu

CONFIDENTIAL
PRELIMINARY

Outline

- Background
- What is ECS?
- ECS Calibration
- Standard test methodology for ECS testing
- Burst capture and dynamic DC
- Splitting testsuits and enable Hidden upload
- Comparison of the two approaches
- Summary and outlook
- References

ADVANTEST

Background

- The testing/calibration of ECS (Embedded Current Sensor) in MDM (Mobile Data Modem) requires to set different load and drive strength to device, measure analog external supply currents along with reading out 10bit digital codes.

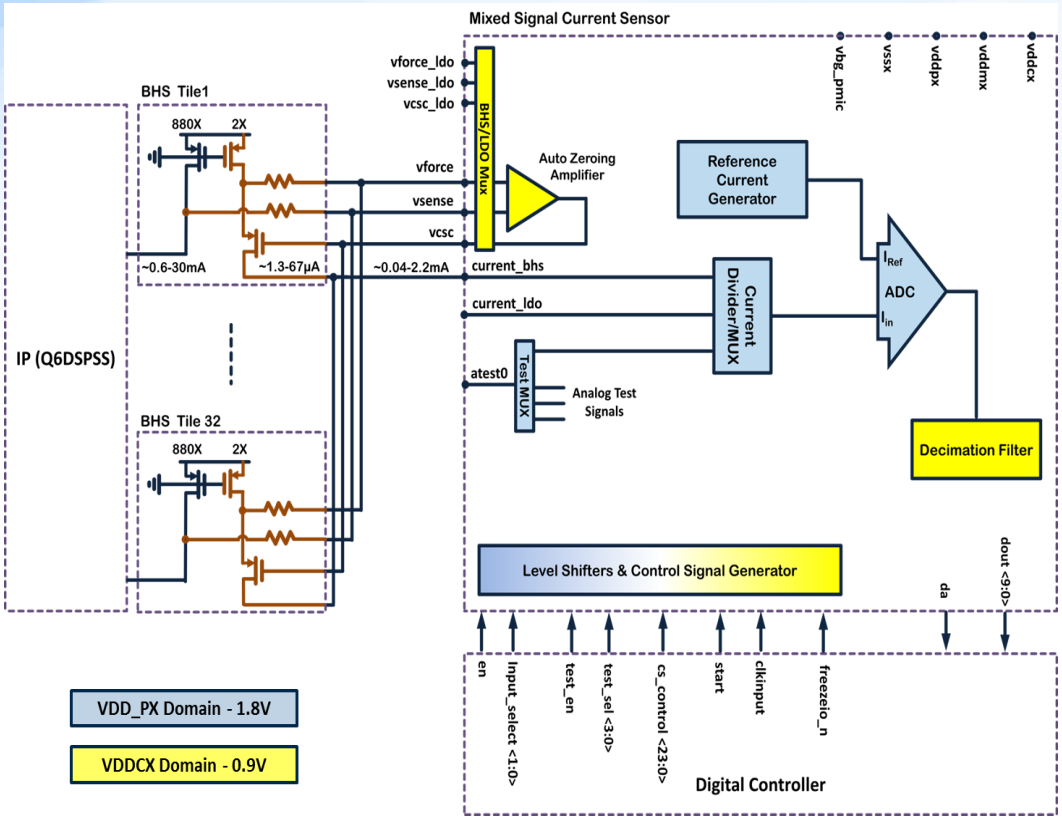
- The test system must apply load-specific pattern and capture the digital codes measure the current consumption accordingly, which will require continuous switching from pattern execution to DC measurements implemented in C++ and back, thus cause a lots of overhead and break time.

- Furthermore, with multi-site configurations, the test time for ECS will extremely increase from single site(~458mS) to x12 sites (~687mS).

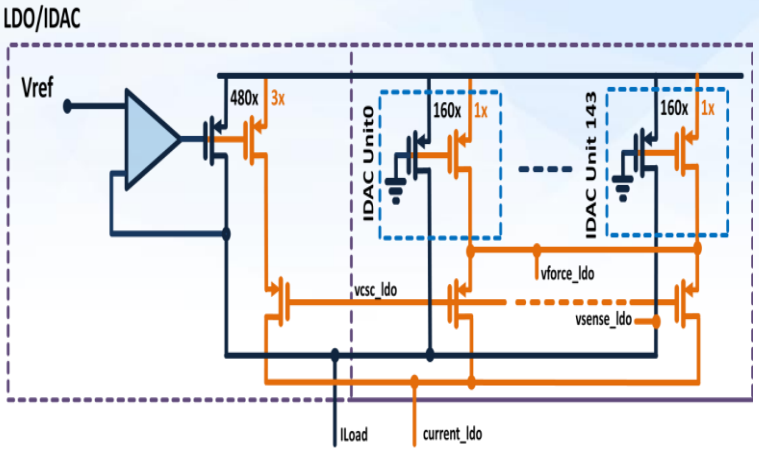
- It's critical to optimize the test methodology and achieve better MSE, thus bring down the over all test time for multi-sites ECS testing.

ADVANTEST

What is ECS?



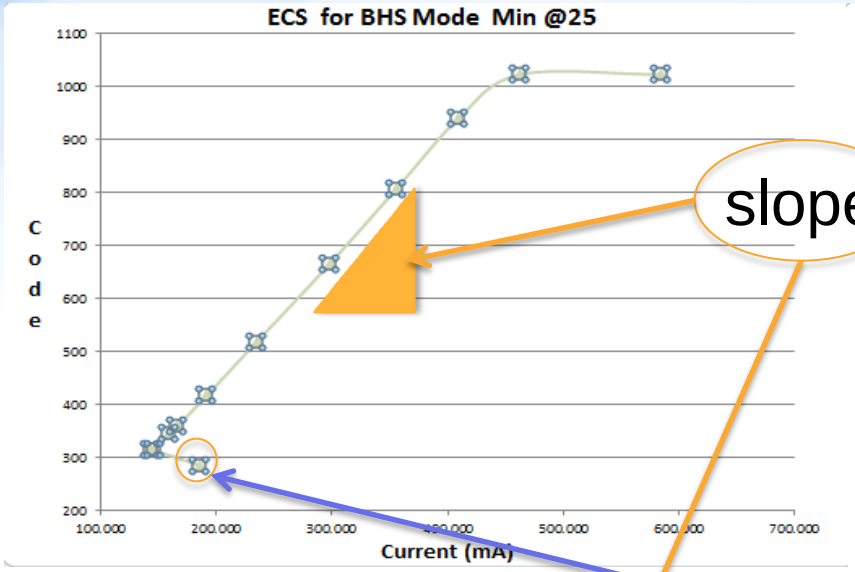
BHS Mode

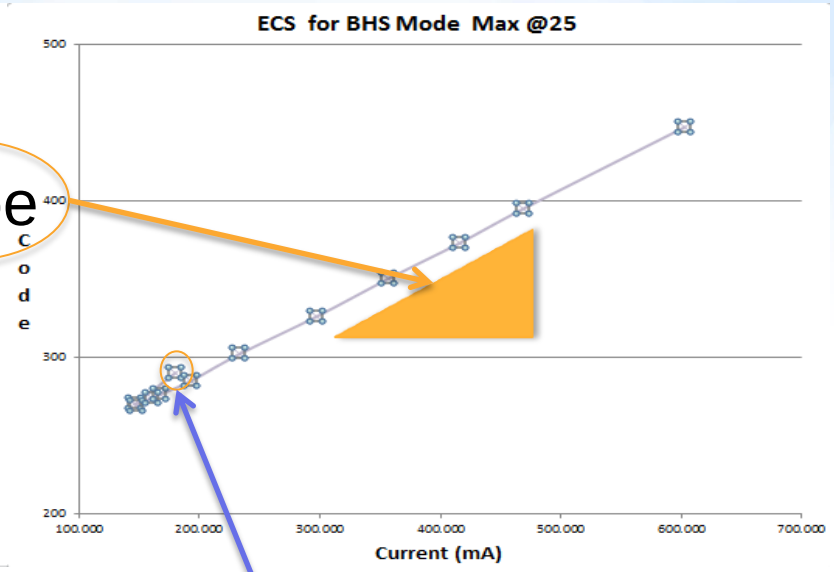


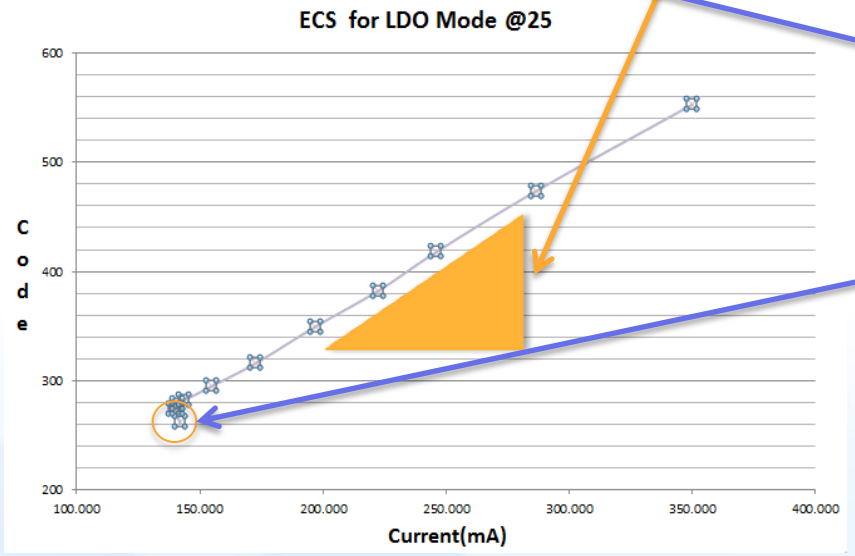
LDO Sensing Mode

ECS is Embedded Current Sensor, which is a new mixed signal IP that has been designed to measure real time actual analog current consumption of DSP block that are powered by either Block Head Switch (BHS) or LDO, thus it can be used to measure the power consumption of individual IPs such as DSP.

ECS Calibration







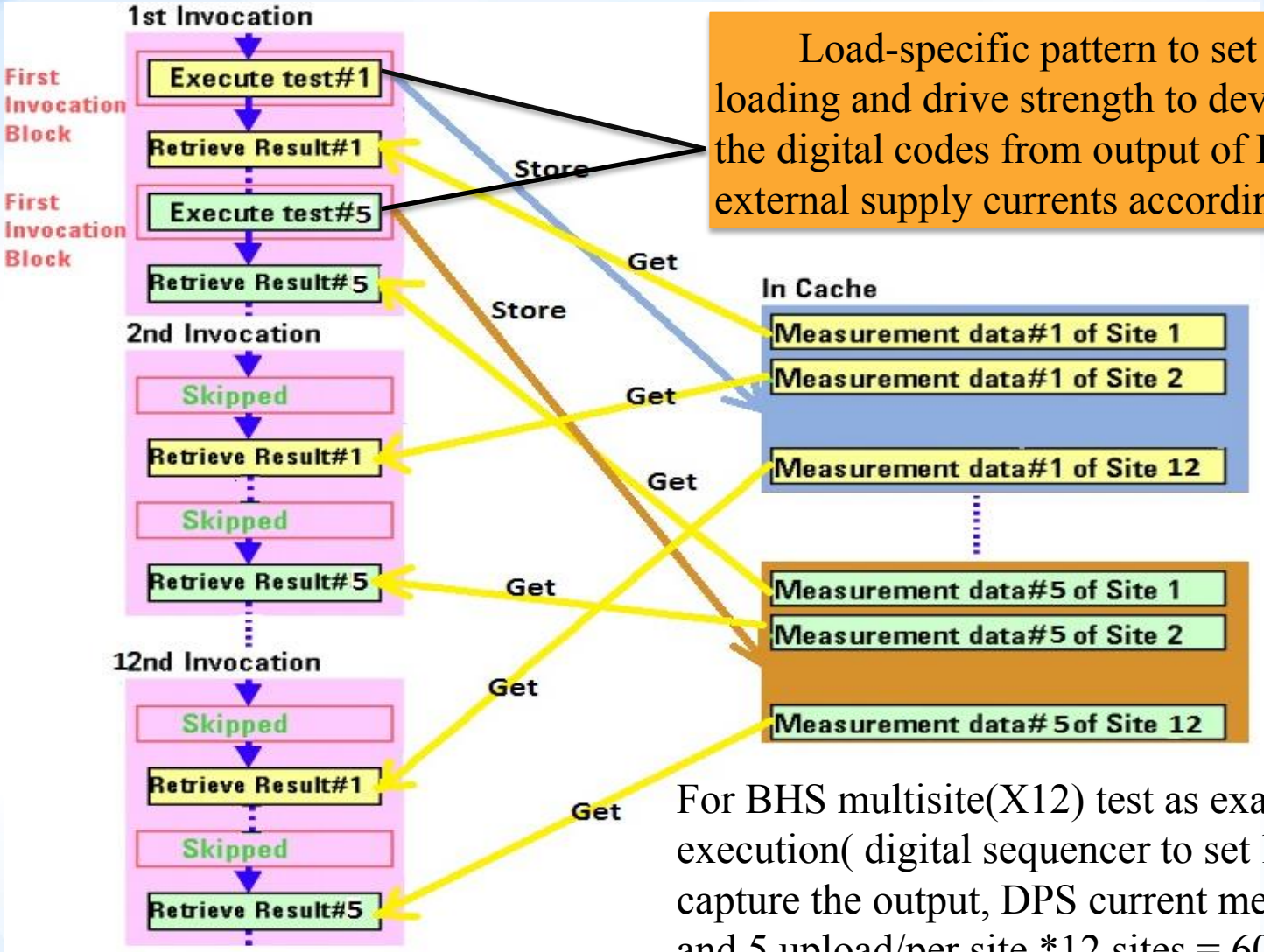
slope

offset

Calibration Goals is to calibrate the slope and offset error, mainly caused by the process temperature variation on reference current as well as other blocks.

ADVANTEST

Standard test methodology for ECS testing

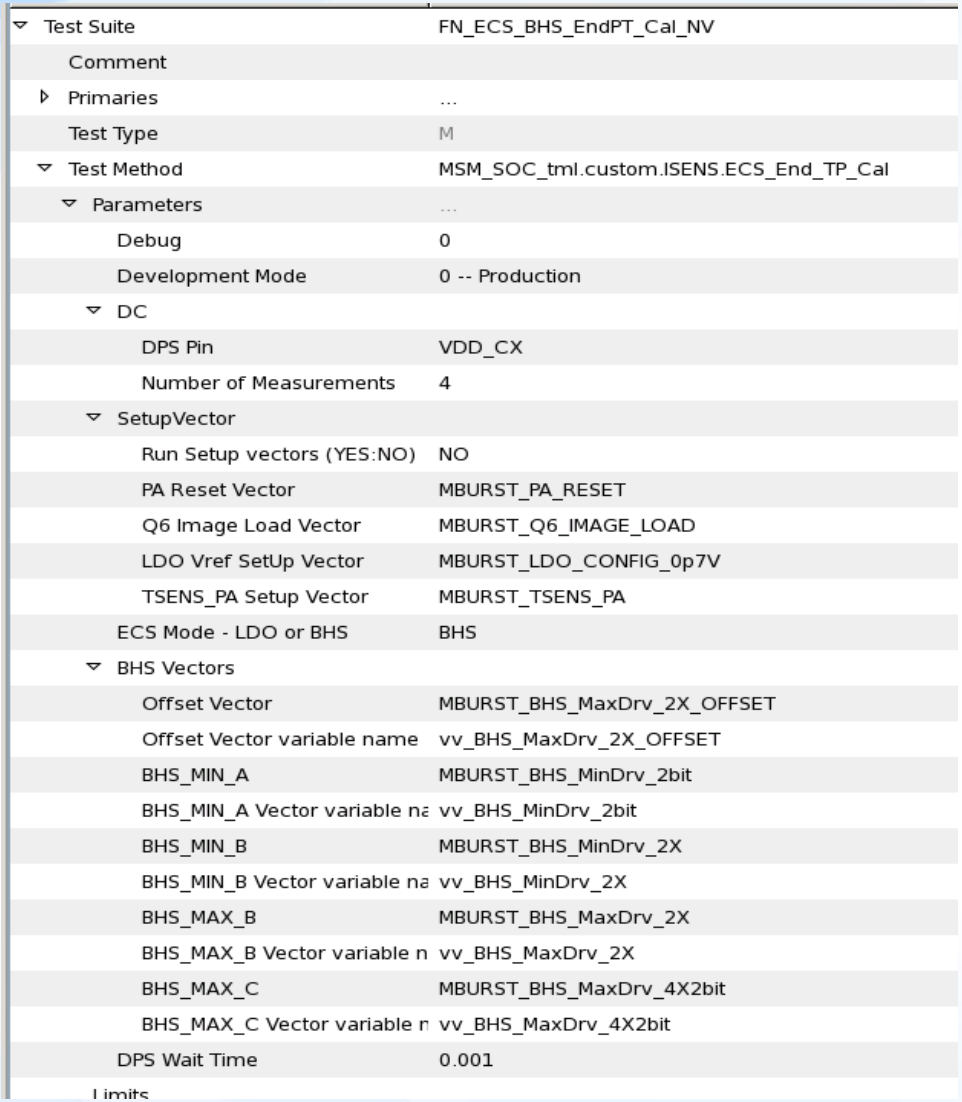


Load-specific pattern to set different loading and drive strength to device, capture the digital codes from output of ECS, measure external supply currents accordingly.

For BHS multisite(X12) test as example: 5 test execution(digital sequencer to set loading and capture the output, DPS current measurement) and 5 upload/per site *12 sites = 60 upload.

ADVANTEST

Standard test methodology for ECS testing



Test Suite	FN_ECS_BHS_EndPT_Cal_NV
Comment	
Primaries	...
Test Type	M
Test Method	MSM_SOC_tml.custom.ISENS.ECS_End_TP_Cal
Parameters	...
Debug	0
Development Mode	0 -- Production
DC	
DPS Pin	VDD_CX
Number of Measurements	4
SetupVector	
Run Setup vectors (YES:NO)	NO
PA Reset Vector	MBURST_PA_RESET
Q6 Image Load Vector	MBURST_Q6_IMAGE_LOAD
LDO Vref SetUp Vector	MBURST_LDO_CONFIG_0p7V
TSENS_PA Setup Vector	MBURST_TSENS_PA
ECS Mode - LDO or BHS	BHS
BHS Vectors	
Offset Vector	MBURST_BHS_MaxDrv_2X_OFFSET
Offset Vector variable name	vv_BHS_MaxDrv_2X_OFFSET
BHS_MIN_A	MBURST_BHS_MinDrv_2bit
BHS_MIN_A Vector variable na	vv_BHS_MinDrv_2bit
BHS_MIN_B	MBURST_BHS_MinDrv_2X
BHS_MIN_B Vector variable na	vv_BHS_MinDrv_2X
BHS_MAX_B	MBURST_BHS_MaxDrv_2X
BHS_MAX_B Vector variable n	vv_BHS_MaxDrv_2X
BHS_MAX_C	MBURST_BHS_MaxDrv_4X2bit
BHS_MAX_C Vector variable n	vv_BHS_MaxDrv_4X2bit
DPS Wait Time	0.001
Limits	

Setup vectors to power up DSP core, BHS vectors to set different load and drive strength:

Offset

Min drive strength – 2 bit load

Min drive strength – 2x load

Max drive strength – 2x load

Max drive strength – 4xplus2bit load

Read out 10bit digital codes of embedded current sensor using Digital Capture and measure analog external supply currents: VDD_CX device power supply.

ADVANTEST

Standard test methodology for ECS testing

Average test time based on test suites [export to csv file] Select MSE formula: MSE = (TestTime(MaxSites)-(MaxSites*TestTime(SingleSite))/(TestTime(SingleSite)*(1-MaxSites)))

seq. #	test suite	# of measure	single site[ms]	multi-sites[ms]	Δ = multi - single[ms]	MSE[%]
483	ECS_Disconnect	5	23.568	24.099	0.531	99.80
484	FN_ECS_PA_reset_NV	5	73.394	73.033	-0.361	100.04
485	FN_ECS_Q6_IMAGE_LOAD_NV	5	14.394	14.670	0.276	99.83
486	FN_ECS_IDDQ_CX_AFT_Q6_Load_FUNC_NV	5	2.933	3.349	0.416	98.71
487	FN_ECS_BHS_EndPT_Cal_NV	5	225.129	366.790	141.661	94.28
488	FN_MBURST_LDO_CONFIG_0p7V	5	2.882	4.234	1.352	95.74
489	FN_ECS_LDO_EndPT_Cal_NV	5	116.040	200.974	84.934	93.35

More than 230mS test time delta is observed between x12 and x1 for ECS testing.

Since the digital captured data upload/DC measurement data and post data processing are handled serially site by site, which explain the test time gap between x1 and x12: more sites enabled, and bigger delta.

Another reason could be continuous switching from pattern execution to DC measurements implemented in C++ and back, which add a lot of overhead/break time.

ADVANTEST

Standard test methodology for ECS testing

Test time details of source codes

Code line	All enabled sites total (ms)	calls	avg (ms)	Code
310	75.783	5	15.157	DIGITAL_CAPTURE_TEST();
311	0.304	5	0.061	WAIT_TEST_DONE();
312				
313				FOR_EACH_SITE_BEGIN_EH
314				{
315	0.005	5	0.001	INT c_site = SITE_IDX;
316				switch(Index)
317				{
318				case 1:
319	1.736	1	1.736	mI_capture_Data1[c_site] = VECTOR(_CaptureParams.CaptureVariableName).getVectors();
320				break;
321				case 2:
322	1.659	1	1.659	mI_capture_Data2[c_site] = VECTOR(_CaptureParams.CaptureVariableName).getVectors();
323				break;
324				case 3:
325	1.648	1	1.648	mI_capture_Data3[c_site] = VECTOR(_CaptureParams.CaptureVariableName).getVectors();
326				break;
327				case 4:
328	1.671	1	1.671	mI_capture_Data4[c_site] = VECTOR(_CaptureParams.CaptureVariableName).getVectors();
329				break;
330				case 5:
331	1.655	1	1.655	mI_capture_Data5[c_site] = VECTOR(_CaptureParams.CaptureVariableName).getVectors();
332				break;
333				default:
334				cout << "Wrong Capture Index given,Pls double check again..." << endl;
335				break;
336				}
337				}
338				FOR_EACH_SITE_END_EH
339				
340				
341				
342	15.012	5	3.002	WAIT_TIME(3 ms);
343				
344				*// The vector is still the (L2+Core) component vector at this point.*
345				DPS_TASK dpsTask;
346	0.060	5	0.012	dpsTask.pin(_VDDPIN).min(0 A).max(1.5 A).measurementMode(TM::MEASURE_CURRENT);
347	0.005	5	0.001	dpsTask.wait(_dpsWaitTime ms);
348	0.007	5	0.001	dpsTask.samples(10).execMode(TM::PVAL);
349	0.004	5	0.001	dpsTask.trigMode(TM::INTERNAL);
350	6.529	5	1.306	dpsTask.execute();

Use TP360 test time breakdown analyzer to check most time consuming code and test actions.

From left test time report for single site, we can observe average 1.67 mS upload time for each capture, ideally we can enable SMC/hidden upload to put all the upload in the background, which can help to save test time for 12 site :

8 captures (BHS 5caps, LDO 3caps)
X 12 sites X 1.67 mS = 160.32 mS

ADVANTEST

Standard test methodology for ECS testing

Test time details of source codes

Code line	All enabled sites			Code
	total(ms)	calls	avg(ms)	
61				
62				/// compute() is called for each site
63				virtual void compute()
64				{
65				// call parent
66	0.163	1	0.163	ECS_EndPTCal::compute();
67				
68				// or put your code here
69				}
70				
71				/// datalog() is called for each site
72				virtual void datalog()
73				{
74				// call parent
75	0.578	1	0.578	ECS_EndPTCal::datalog();
76				
77				// or put your code here
78				}

Test time report also shows:
calculation time = 0.163mS/per site
data log time = 0.578mS/per site

Since these post processing: calculation and data logging are handled serially site by site, test time estimation for 12 sites:
*calculation time = 0.163*12 =1.956mS*
*data log time = 0.578*12 =6.936mS*

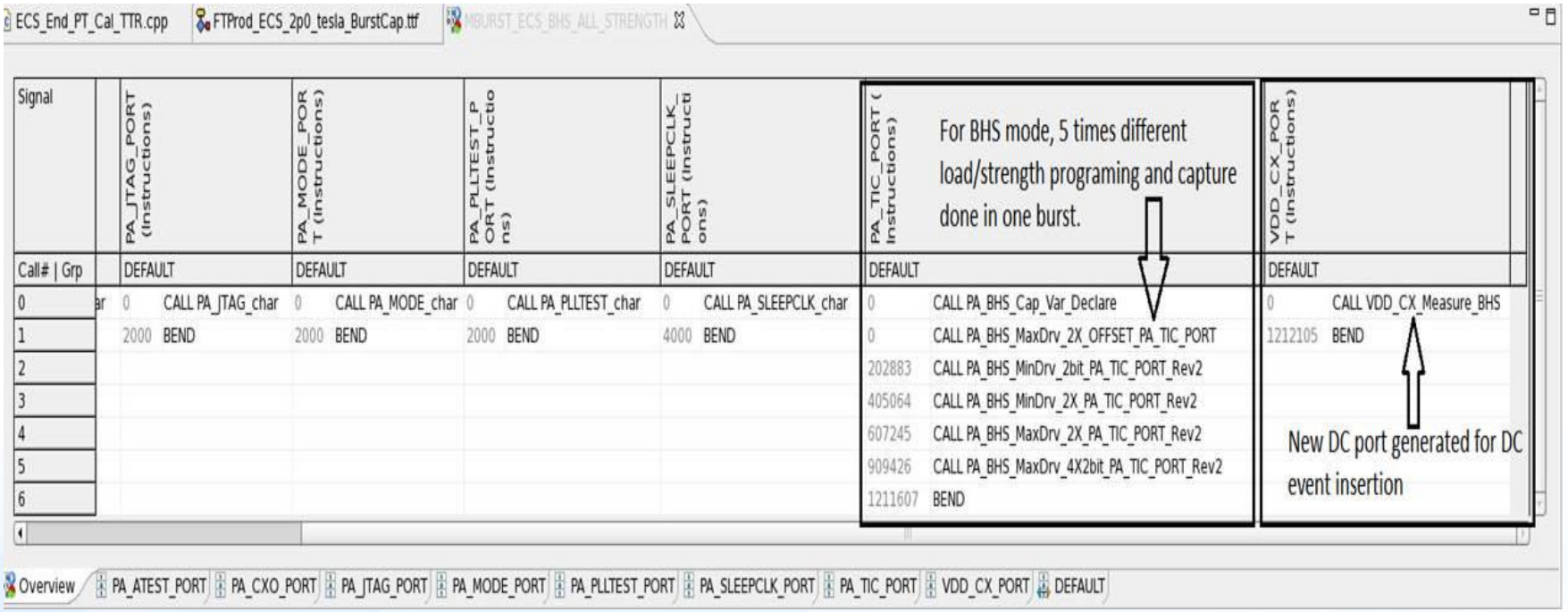
Totally post processing time: 8.892mS

ADVANTEST

Burst capture and dynamic DC

Smartest allow to do burst capture and execute DC actions from digital sequencer.

Created burst capture together with DPS current measurement from digital pattern for BHS and LDO in order to optimize test time, Accordingly new testmethod ECS_End_PT_Cal_TTR_TM (TTR version) was generated. Here is BHS test for example:

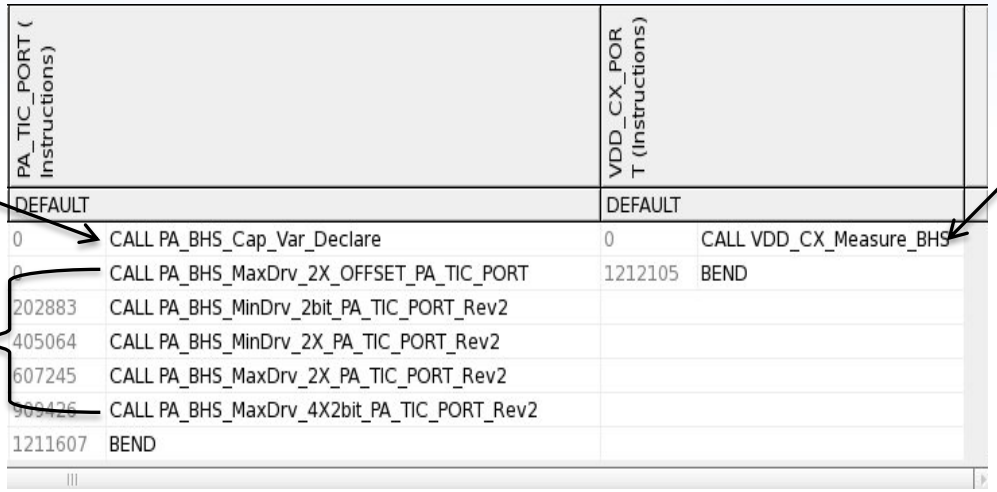


Burst capture and dynamic DC

Burst capture: set up digital capture on a burst pattern with multiple main patterns that digital capture is done in them.

Define the Vector Variable to accommodate for all the digital captures in main patterns.

Digital capture was done in each main patterns.



PA_TIC_PORT (Instructions)		VDD_CX_POR T (Instructions)	
DEFAULT		DEFAULT	
0	CALL PA_BHS_Cap_Var_Declare	0	CALL VDD_CX_Measure_BHS
0	CALL PA_BHS_MaxDrv_2X_OFFSET_PA_TIC_PORT	1212105	BEND
202883	CALL PA_BHS_MinDrv_2bit_PA_TIC_PORT_Rev2		
405064	CALL PA_BHS_MinDrv_2X_PA_TIC_PORT_Rev2		
607245	CALL PA_BHS_MaxDrv_2X_PA_TIC_PORT_Rev2		
909426	CALL PA_BHS_MaxDrv_4X2bit_PA_TIC_PORT_Rev2		
1211607	BEND		

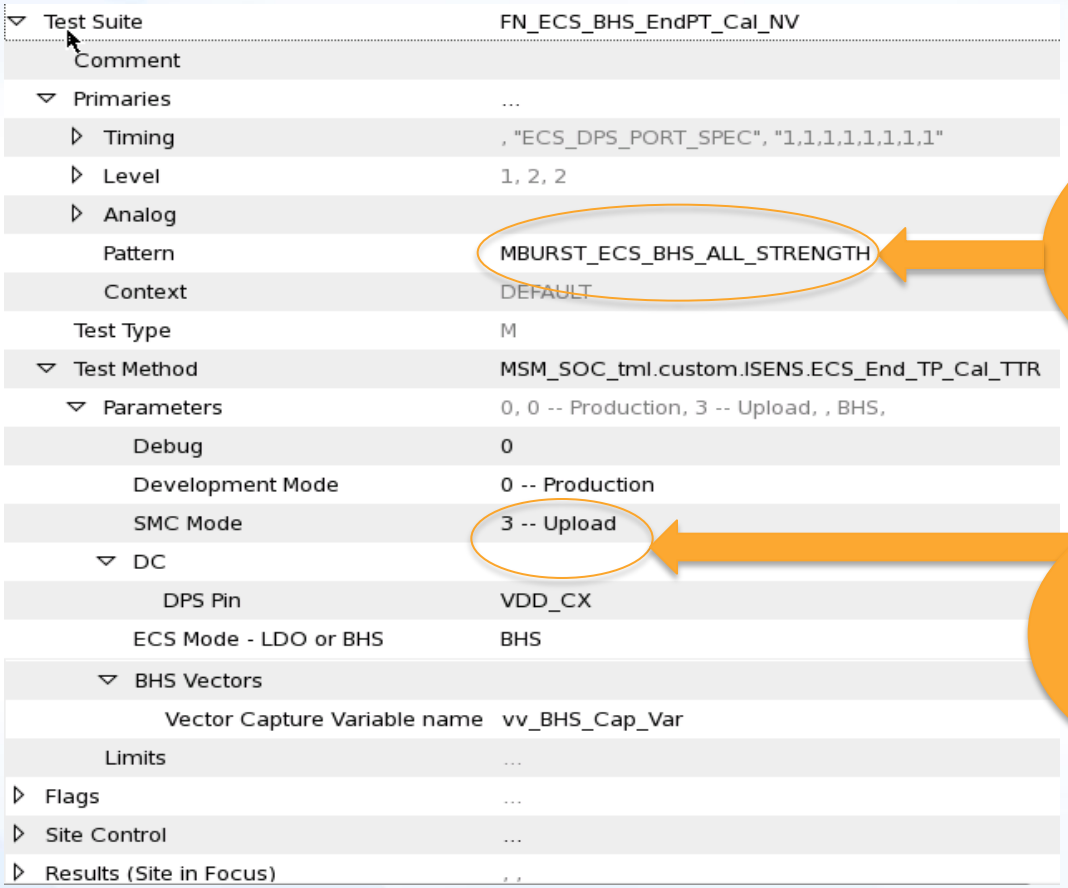
Add DC port pattern and insert DC events between digital capture patterns, do DC measurements at properly position synchronized with digital patterns.

Dynamic DC: as new set of features on DC Scale cards, which allow DC measurements controlled by, exactly synchronized with digital patterns.

ADVANTEST

Burst capture and dynamic DC

The new test methodology can ease the test method coding /complicate data handling and allow to easily apply existing Hidden upload feature of SmartCalc framework for further test time reduction.



Burst capture and execution DC measurement from digital pattern

Easily apply existing SmartCalc/Hidden upload framework, as one result for all digital capture.

Burst capture and dynamic DC

Burst capture and execution DC actions from digital sequencer both can benefit test case with cutting overhead and break time. 47.793mS test time saved for single site.

Test Program 360 - Test Time Comparison Report

Comparison report - overall test time			
Compare item	**R1:report as reference**	**R2:report to compare**	**Comparison result**
Measure name	Single site	Single site	
Test time per flow(ms)	324.571	276.778	47.793ms or 14.72% improved
Test time per device(ms)	324.571	276.778	47.793ms or 14.72% improved
MSE(%)			
Test time data	source test time report	target test time report	comparison result
Measure conditions	source measure conditions	target measure conditions	comparison result

Average test time based on test suites [export to csv file]

Select MSE formula: MSE = (Tn-N*T1)/(T1*(1-N))

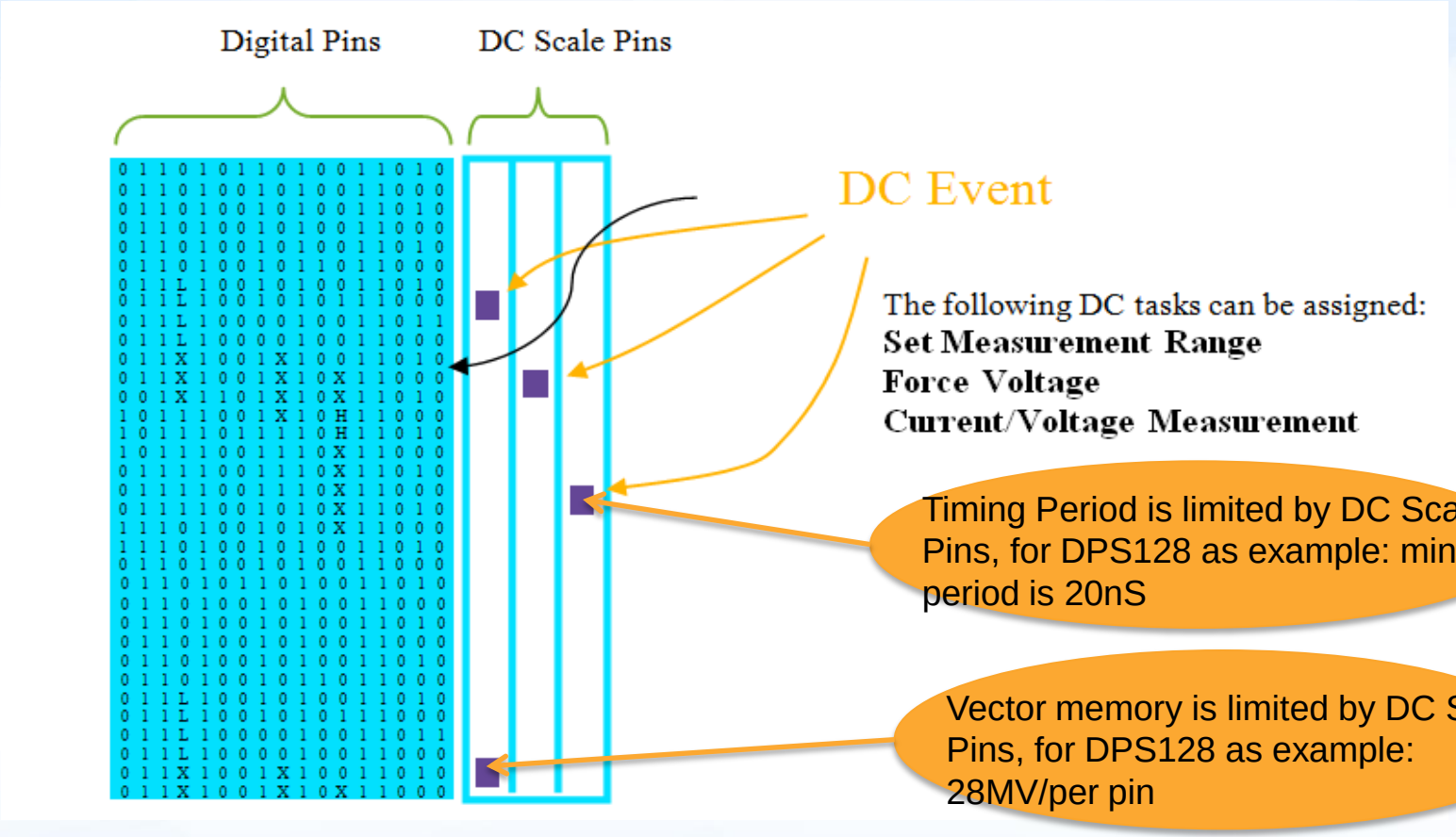
Test suite	# of measure(R1/R2)	Org testmethod R1[ms]	Burst Capture R2[ms]	impr.[ms]	impr.%
FW_Setup	20/20	1.280	1.148	0.132	10.31
DCcont_neg_NV	20/20	14.367	14.787	-0.420	-2.92
FN_ECS_PA_reset_NV_2_3	20/20	70.298	69.848	0.450	0.64
FN_ECS_Q6_IMAGE_LOAD_NV_2_3	20/20	12.804	12.838	-0.034	-0.27
FN_ECS_IDDQ_CX_AFT_Q6_Load_FUNC_NV_2_3	20/20	2.435	2.395	0.040	1.64
FN_ECS_BHS_EndPT_Cal_NV	20/20	128.308	93.748	34.560	26.94
FN_MBURST_LDO_CONFIG_0p7V_3	20/20	2.491	3.441	-0.950	-38.14
FN_ECS_LDO_EndPT_Cal_NV	20/20	79.408	64.622	14.786	18.62

X12 test time go down from original 687mS to 340ms.

ADVANTEST

Burst capture and dynamic DC

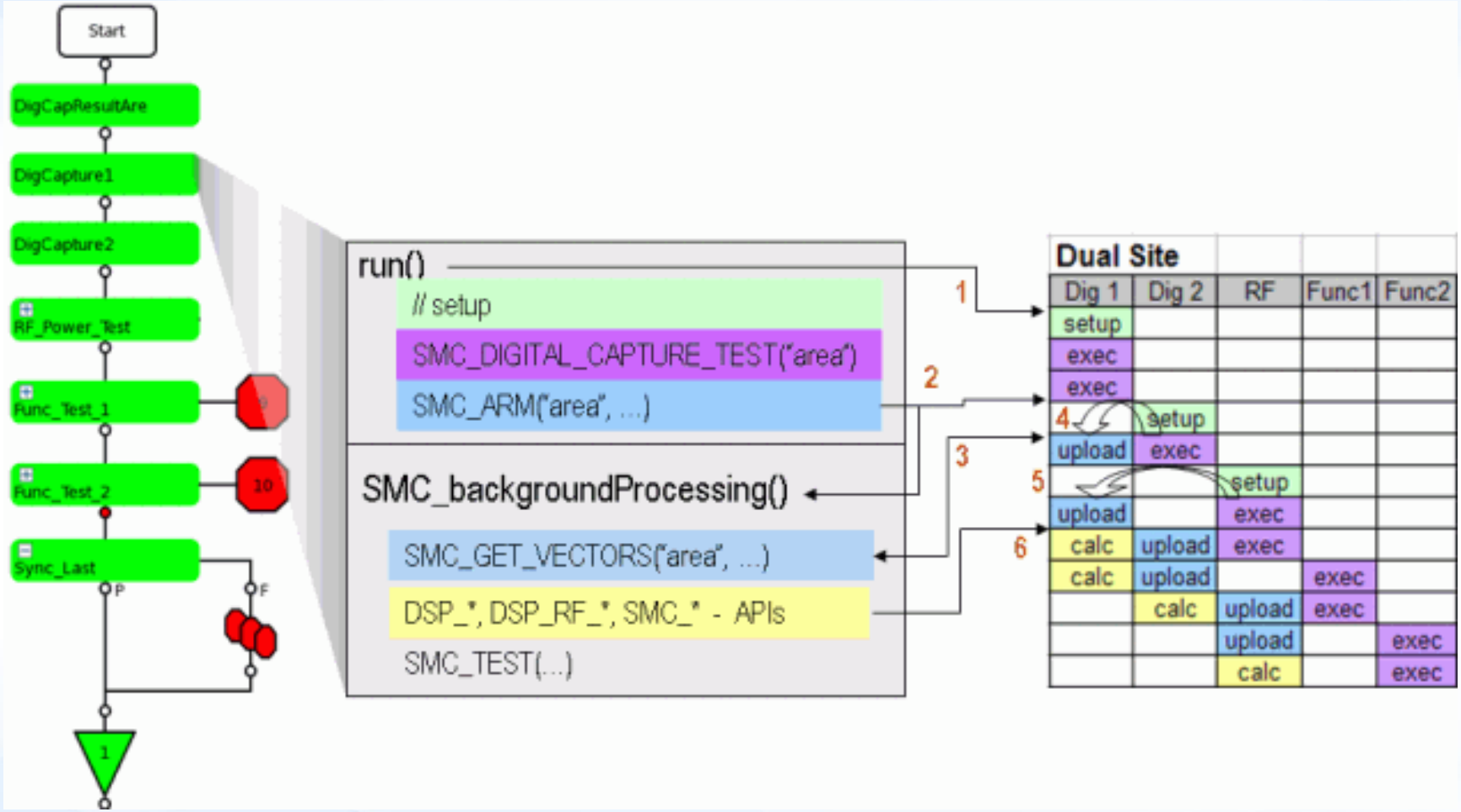
Things to be noted with Dynamic DC measurement:



Always develop your test program set up with Ports concept.

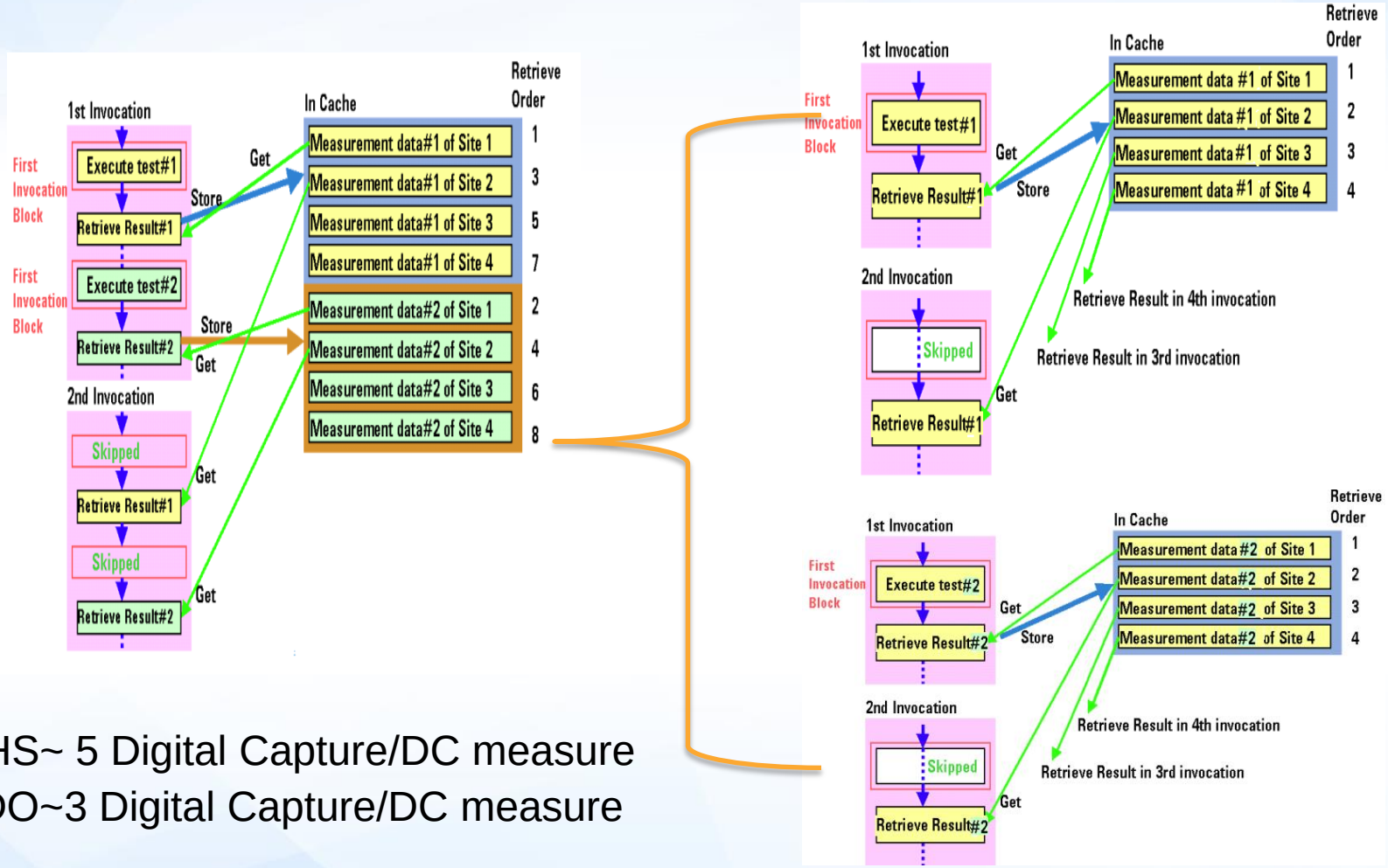
Splitting testsuits and enable Hidden upload

SmartCalc provides an implementation framework to handle hidden digital and analog upload and multithreaded calculation.



Splitting testsuits and enable Hidden upload

Split one complex testsuit into multiple simple testsuits, it allows to easily implement existing SmartCalc framework without further exploration.

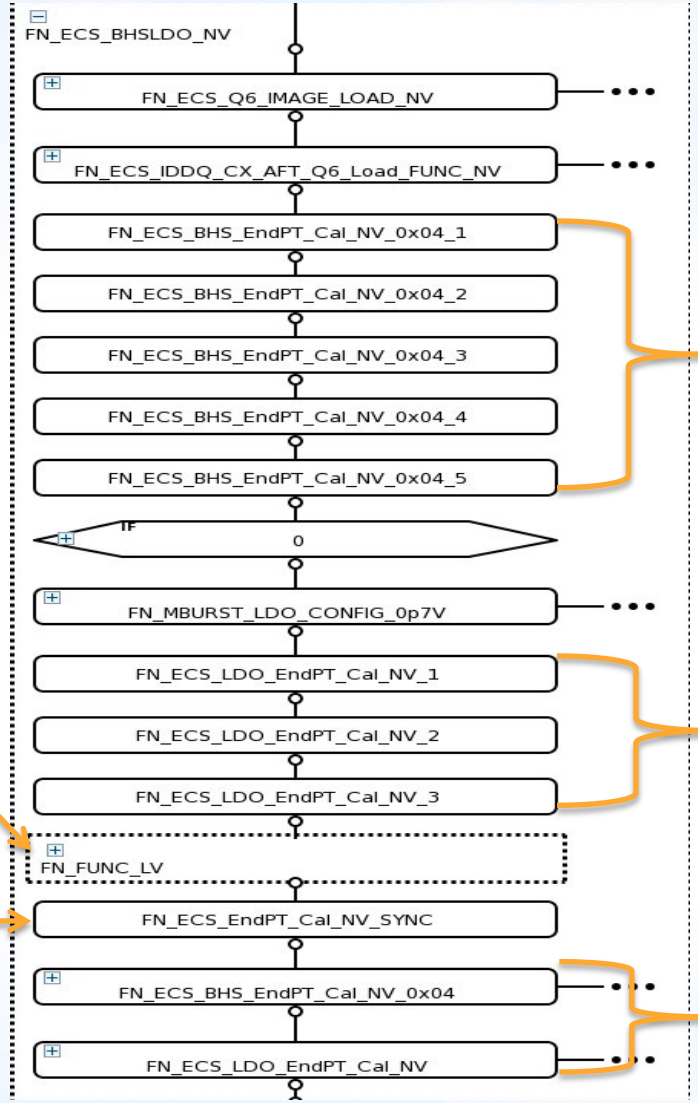


BHS~ 5 Digital Capture/DC measure

LDO~3 Digital Capture/DC measure

ADVANTEST

Splitting testsuits and enable Hidden upload



SMC_backgroundProcessing()

SMC_GET_VECTORS("area", ...)

When running functional testing, trigger the upload in the background.

SMC_SYNCHRONIZE()

BHS test: Load-specific pattern to set 5 different load and drive strength, and digital Capture/ DPS current measurement accordingly

LDO test: Load-specific pattern to set 3 different load and drive strength, and digital Capture/ DPS current measurement accordingly

Perform calculation of slope/offset and the data logging and binning.

ADVANTEST

Splitting testsuits and enable Hidden upload

Test time detail collected for 12 Sites with turning on Hidden upload feature:

Sites	TestSuiteName	Count	Sum	Min	Max	Mean
12	FN_ECS_Q6_IMAGE_LOAD_NV	20	735.001	36.28	37.498	36.7501
12	FN_ECS_IDDQ_CX_AFT_Q6_Load_FUNC_NV	20	59.192	2.553	4.26	2.9596
12	FN_ECS_BHS_EndPT_Cal_NV_0x04_1	20	726.31	35.515	36.908	36.3155
12	FN_ECS_BHS_EndPT_Cal_NV_0x04_2	20	697.838	34.44	36.125	34.8919
12	FN_ECS_BHS_EndPT_Cal_NV_0x04_3	20	693.955	34.232	38.076	34.6977
12	FN_ECS_BHS_EndPT_Cal_NV_0x04_4	20	694.051	34.262	36.063	34.7025
12	FN_ECS_BHS_EndPT_Cal_NV_0x04_5	20	694.203	34.284	35.739	34.7102
12	FN_MBURST_LDO_CONFIG_0p7V	20	98.166	4.668	5.573	4.9083
12	FN_ECS_LDO_EndPT_Cal_NV_1	20	850.319	42.204	43.897	42.5159
12	FN_ECS_LDO_EndPT_Cal_NV_2	20	697.228	34.525	36.248	34.8614
12	FN_ECS_LDO_EndPT_Cal_NV_3	20	699.704	34.745	35.94	34.9852
12	FN_ECS_EndPT_Cal_NV_SYNC	20	47.977	1.825	11.627	2.39885
12	FN_ECS_BHS_EndPT_Cal_NV_0x04	20	185.995	2.23	11.726	9.29975
12	FN_ECS_LDO_EndPT_Cal_NV	20	195.295	7.378	12.29	9.76475
					Total:	353.7617

Since capture post processing: data upload can be moved in the background, it can significantly improves MSE, thus reduce test time for x12 sites from original 687mS to 353ms.

ADVANTEST

Comparison of the two approaches

- Pro's and con's

Burst capture and dynamic DC	Splitting testsuits and enable Hidden upload
+ Ease the test method coding.	+ Ease the test method coding.
+ Avoid complicated data handling.	+ Avoid complicate data handling.
+ Allow to easily apply existing Hidden upload feature of SmartCalc framework.	+ Allow to easily apply existing Hidden upload feature of SmartCalc framework.
+ Better TTR results as with cutting a lots of overhead /break time and hidden upload.	+ No change on the test program setup: pin configuration, level/timing/vector setup.
- Change a lot on Test program setup: add DPS port , generate timing for DPS port, and create new burst capture pattern.	- Create more testsuits which cause more overhead from running testsuits to testsuits.
- Need to insert DPS port into digital sequencer and add DC event at properly position.	- New test flow is more complicated , need to well arrange the testsuits running sequence.
- Introduce limitations to digital sequencer timing period and vector memory.	- Can not cut the overhead and break time but increase new overhead.

ADVANTEST

Comparison of the two approaches

- Test time comparison

	Burst capture and dynamic DC	Splitting testsuits and enable Hidden upload
Single Site	276 mS	287 mS
Multi(X12) Site	340 mS	353 mS

Burst capture and dynamic DC: it cuts overhead / break time and hide upload time, can achieve better and lower test time, which involve big changing on test program setup: turn on test point in .technology, add new DPS port , generate timing for DPS port, and create new burst capture pattern with DPS port.

Splitting testsuits and enable Hidden upload: make use of hidden upload feature of SmartCalc framework, which can hide upload time and it's easy to implement without any change on the test program setup: pin configuration, level/timing/vector setup. But can not cut the overhead and break time.

ADVANTEST

Summary and outlook

- This paper introduced two solutions to TTR for ECS Calibration and discussed pros and cons of the different solutions.

- Different TTR technique/skill are discussed: hidden upload feature of the SmartCalc framework has been implemented in order to provide optimized full automatic upload capabilities behind a currently running test (sequencer execution).

- Burst Capture (set up digital capture on a burst pattern with multiple main patterns that digital capture is done in them) is used to cut the overhead/break time between multiple digital capture testing.

- Dynamic DC, as new set of features on DC Scale cards, which allow DC measurements controlled by, exactly synchronized with digital patterns, thus cut the overhead/break time between digital sequencer and testmethod.

ADVANTEST

Summary and outlook

- There is some potential improvements that can be applied to the solution: Splitting testsuits and enable Hidden upload , Re-write the *SMC_backgroundProcessing()* and allow it to handle multiple different *SMC_ARM()* call within one testsuits, So that we don't need to split the original testsuits and this will also further reduce the overhead.

- SmartRDI (Rapid Development Interface) provides a high level of programming interface to create test method program. It is a new V93000 API set to provide automated sequencer based pattern burst generation for DC test, digital capture. That can be applied to the solution: Burst capture and dynamic DC, ease the test programming.

ADVANTEST

References

- Advantest V93000 TDC Documentation.
  - Topic 42637 Multisite Test Method Programming
  - Topic 42640 Measurement Data Management
  - Topic 119874 How the SmartCalc framework handles hidden upload
  - Topic 113758 How to set up a dynamic DC test
  - Topic 142069 SmartRDI Introduction.

- Test Program 360 Cookbook.

- Issues with Digital Capture implementation, TCE User Group Sharing.

ADVANTEST

THANK YOU!

ADVANTEST