AnalogSet_Generation_Issue with SOC_tml

Prepared By: Stephen Fu

CONFIDENTIAL

PRELIMINARY

Analog resource assignment

ADVANTEST

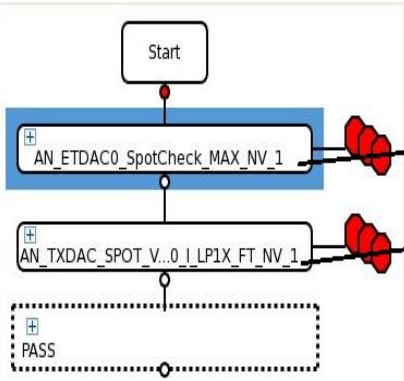
2 Analog pins (etdac_iq & txdac_iq) are connecting to the same digitizer units with different pogo pin: (*X*+ or *X*-) *and* (*XX*+ or *XX*-).

	Name	No	Type	HW	Comment	Tester Channel													
						Site 1			Site 2			Site 3			Site 4			Site 5	
						Pogo	Pad	Unit	Pogo	Pad ▵	Unit	Pogo	Pad	Unit	Pogo	Pad	Unit	Pogo	Pa
1	etdac_iq-		o	MCE		23101	S1	8	23105	S1	6	23107	S1	5	22805	S1	6	22801	S1
2	etdac_iq+		o	MCE		23102	S1	8	23106	S1	6	23108	S1	5	22806	S1	6	22802	S1
3	txdac_iq-		o	MCE		22815	S1	5	22809	S1	8	23109	S1	8	22811	S2	3	22813	S2
4	txdac_iq+		o	MCE		22816	S1	5	22810	S1	8	23110	S1	8	22812	S2	3	22814	S2
5	bbrx_iq-		i	MCE		23113	S2	2	23113	S2	2	23113	S2	2	23113	S2	2	23113	S2
6	bbrx_iq+		i	MCE		23114	S2	2	23114	S2	2	23114	S2	2	23114	S2	2	23114	S2
7	gpsadc_iq+		i	MCE		23116	S2	1	23116	S2	1	23116	S2	1	23116	S2	1	23116	S2

For example, site 1 etdac_iq share the same digitizer unit 8 with site 3 txdac_iq.

There are 2 similar test (SpotCheck) for ETDAC and TXDAC, which are using the same digitizer unit, they will need to create different sequencer for ETDAC and TXDAC, currently SOC_tml cannot differentiate it, it will generate the same sequencer for both, which cause problem during production running.

TestSuits



type filter text

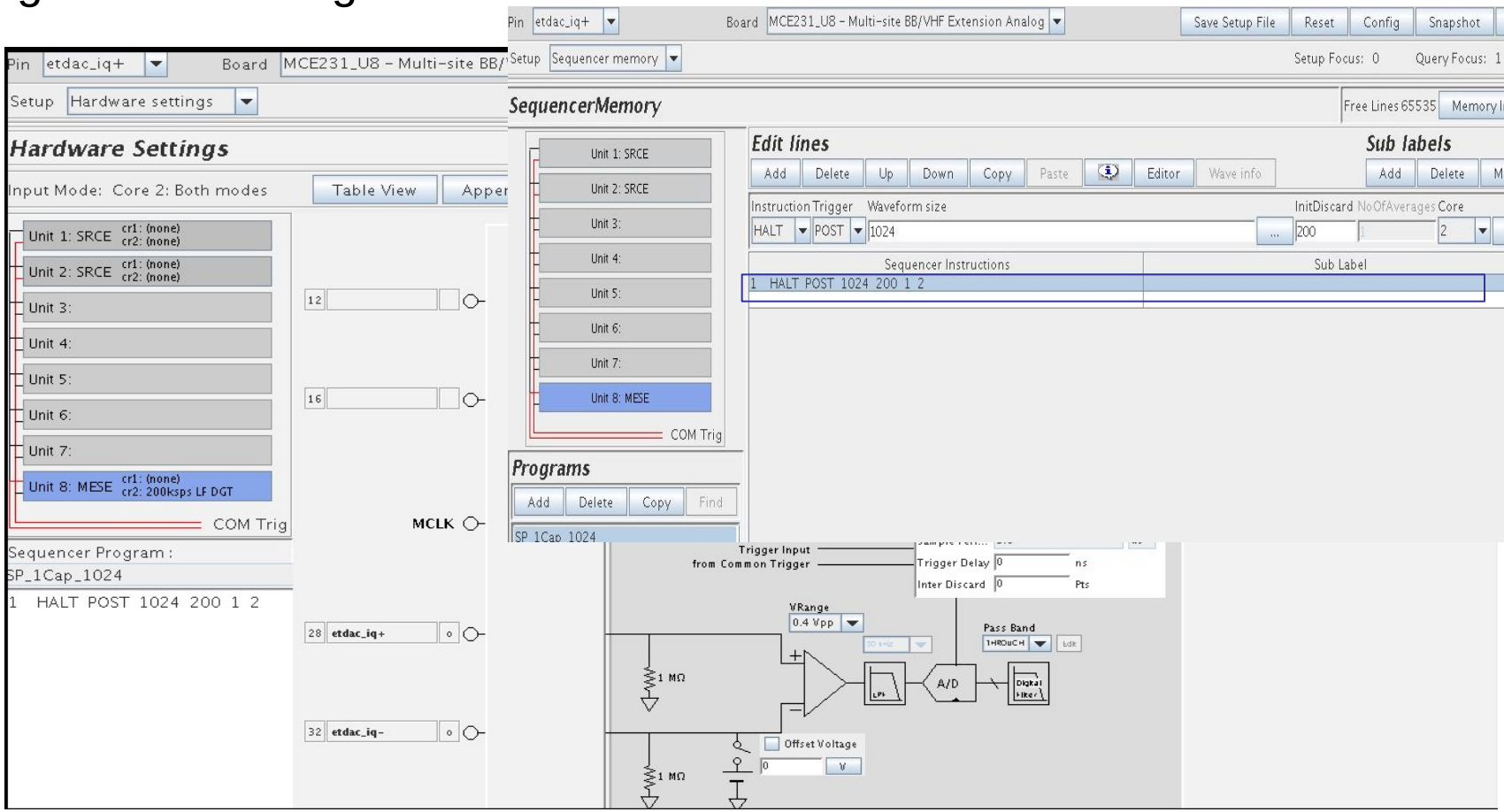
Property	Value
Test Suite	AN_ETDAC0_SpotCheck_MAX_NV_1
Comment	
Primaries	...
Test Type	M
Test Method	MSM_SOC_tml.custom.DAC.SpotCheck
Parameters	...
Debug	3
Development Mode	3 -- Generate ONLY
Measurement Mode	DIG
SMC Mode	0 -- OFF
DGT	
Number of Channels	1
Pin	etdac_iq+
Connection Type	DIFFERENTIAL
Core Function	LF
Sample Rate	200 KHz
Capture Size	1024
Initial Discard	200
HW Averages	1
Range	0.4
Filter	THRU
Impedance	1M
DC Offset	0
SPOTCHECK_PARAMETER	
Save the DC value	YES
Global Var Name	ETDAC_SPOT
Limits	...

type filter text

Property	Value
Test Suite	AN_TXDAC_SPOT_VMAX_MEAS_DAC0_I_LP1X_F
Comment	
Primaries	...
Test Type	M
Test Method	MSM_SOC_tml.custom.DAC.SpotCheck
Parameters	...
Debug	3
Development Mode	3 -- Generate ONLY
Measurement Mode	DIG
SMC Mode	0 -- OFF
DGT	
Number of Channels	1
Pin	txdac_iq+
Connection Type	DIFFERENTIAL
Core Function	LF
Sample Rate	200 KHz
Capture Size	1024
Initial Discard	200
HW Averages	1
Range	0.4
Filter	THRU
Impedance	1M
DC Offset	0
SPOTCHECK_PARAMETER	
Save the DC value	YES
Global Var Name	TXDAC0_SPOTMAX_I
Limits	...

Test Setup

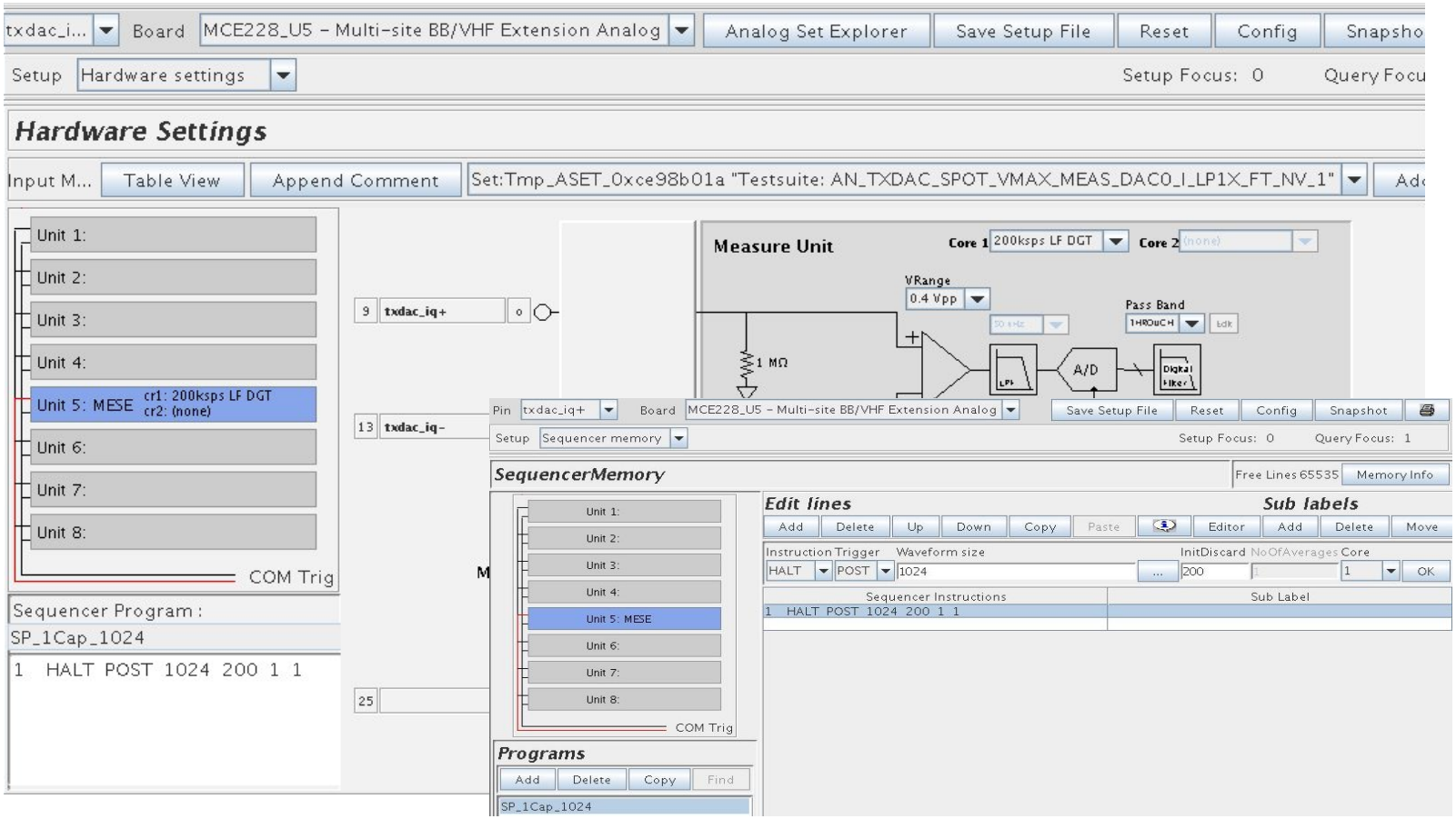
Running testsuit 1: AN_ETDAC0 in development mode 3, it will generate analog set for testsuit 1:



Sequencer program: SP_1Cap_1024 (1 HALT POST 1024 200 1 2) is generated using core 2.

Test Setup

Running testsuit 2: AN_TXDAC in development mode 3, it will generate analog set for testsuit 2:



Sequencer program: SP_1Cap_1024 (1 HALT POST 1024 200 1 1) is generated using core 1.

Problem Statement

Testsuits 1 Sequencer program SP_1Cap_1024: (1 HALT POST 1024 200 1 2)
Testsuits 2 Sequencer program: SP_1Cap_1024 (1 HALT POST 1024 200 1 1)

The sequencer program SP_1Cap_1024 generated by testsuits 2 will overwrite sequencer program generated by testsuits 1 since they are using the same sequencer name: SP_1Cap_1024.

When we switch to production mode,Testsuits 1 will encounter program with getwaveform function due to mismatch waveform ID. However testsuits 2 is running fine without such kind of error.

Check the analog set, testsuit 1 and 2 now share the same sequencer program: SP_1Cap_1024 (1 HALT POST 1024 200 1 1), which can explain above behavior.

Problem Statement

ADVANTEST

Need to change the following functions in SOC_tml/Analog/src/DGT_Settings.cpp

```
STRING DigitizerSettings::getSequencerLabel()
{
    if(!mSequencerLabel.empty()) return mSequencerLabel;
    INT averaging = mSequencerProgram.getMaxAveraging();
    if(averaging <= 1)
    {
        // no averaging
        return "SP_" + String(mNumCaptures) + "Cap_" + String(mSequencerProgram.rawDataSize());
    }

    return "SP_" + String(mNumCaptures) + "Cap_" + String(mSequencerProgram.rawDataSize()) + "_Averaging_" + String(averaging);
}
```

Can put more info in the sequencerlabel to differentiate the sequencer program?

Currently we need to manually change the sequencer label after its generation to avoid overwritten by following testsuits.

End of Slides