THE GTSOC: BLACK BOX CONTROL INTEGRATION WITH THE RTDS SIMULATOR

CHRISTIAN JEGUES, P.ENG

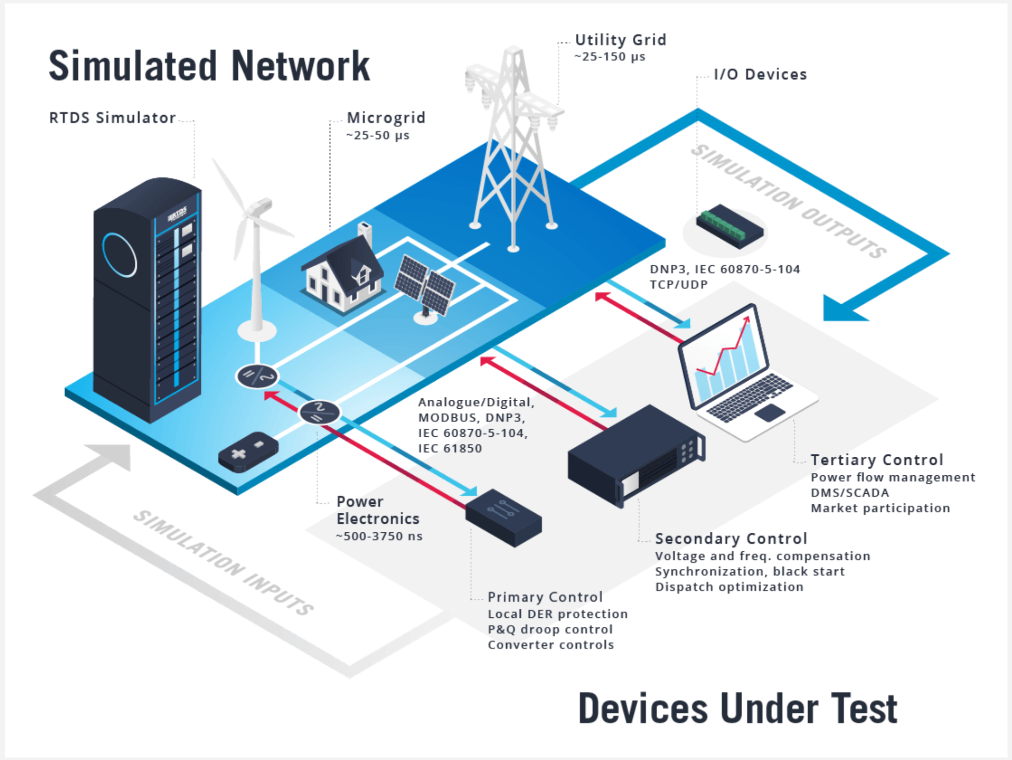
RTDS TECHNOLOGES INC.

Outline

- Why are Black Box Controls Important?
- Black Box Controls with the RTDS
- Introduction to GTSOC
- Deploying Blackbox Controls on the GTSOC
- GTSOC Examples

Why are Black Box Controls Important?

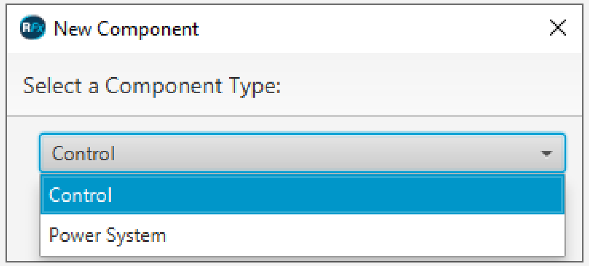
- Controller circuitry may be tightly integrated with power circuitry
- Difficult to scale if a large number of controllers are involved
- Customers want models that accurately reflect the real control and protection equipment provided by vendors
- The same code base used by vendors can be used to create the black box model
- Vendor's IP must be protected
- Generic models have their limitations
- Tuning generic models can be very time consuming

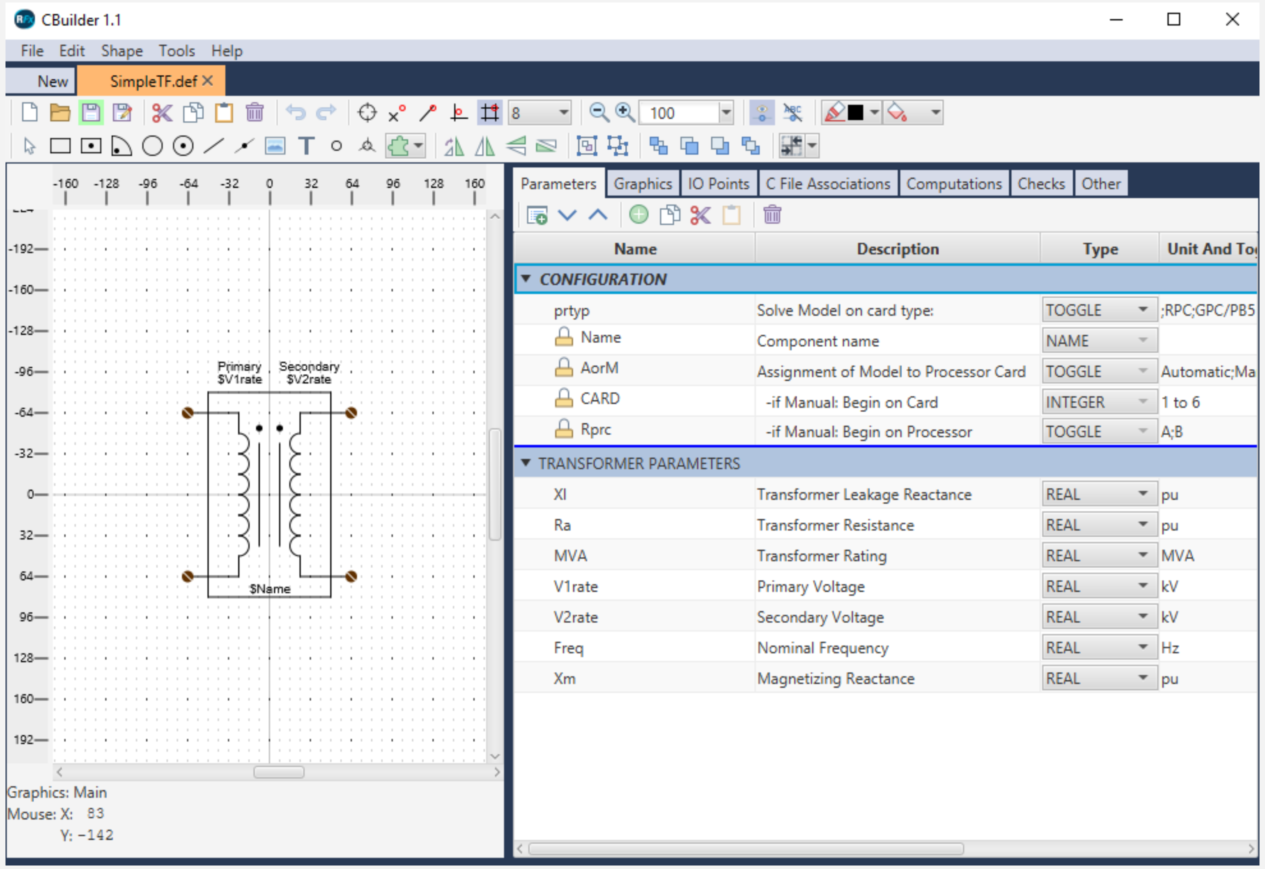


Black Box Controls with the RTDS

Component Builder (Cbuilder)

- Allows users to develop their own component models which would run in real time
- Programmed using subset of C, specific structure required
- May require vendors to maintain two separate code bases



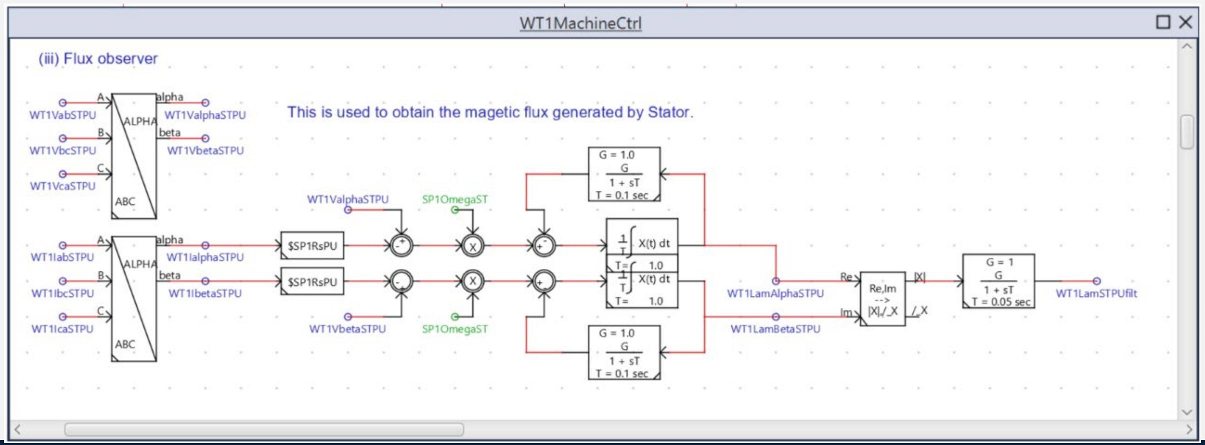


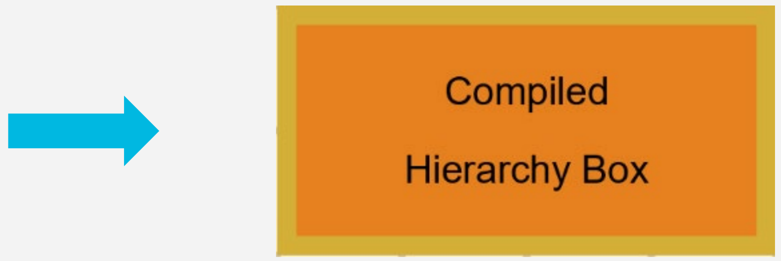
Name	Description	Type	Unit And To
CONFIGURATION			
prtyp	Solve Model on card type:	TOGGLE	;RPC;GPC/PB5
Name	Component name	NAME	
AorM	Assignment of Model to Processor Card	TOGGLE	Automatic;Ma
CARD	-if Manual: Begin on Card	INTEGER	1 to 6
Rprc	-if Manual: Begin on Processor	TOGGLE	A;B
TRANSFORMER PARAMETERS			
Xl	Transformer Leakage Reactance	REAL	pu
Ra	Transformer Resistance	REAL	pu
MVA	Transformer Rating	REAL	MVA
V1rate	Primary Voltage	REAL	kV
V2rate	Secondary Voltage	REAL	kV
Freq	Nominal Frequency	REAL	Hz
Xm	Magnetizing Reactance	REAL	pu

Black Box Controls with the RTDS

Compiled Hierarchy Box

- Blackbox controller could be modelled using standard library components and/or CBuilder components
- Hierarchy boxes can be compiled to secure the contents
- Only the compiled code needs to be sent to the customer
- Vendor would need to verify that the model matches the actual controller
- Vendor may need to maintain this model in addition to the original controller source code

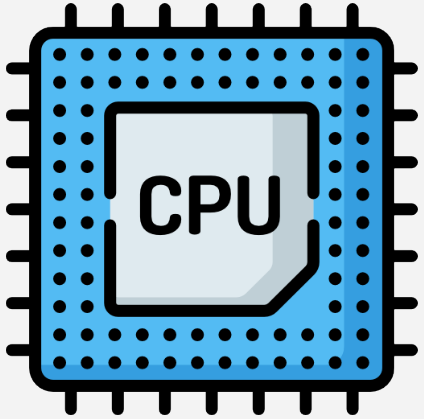




Black Box Controls with the RTDS

Other Hardware

- External PC
  - Windows OS
    - Dynamic Library *.dll
    - Static Library *.lib
  - Linux
    - Shared Library *.so
    - Static Library *.a
- ARMv8-A Processor
  - Supports Bare Metal Execution of Static Library *.a
- Real-time operation is a critical requirement



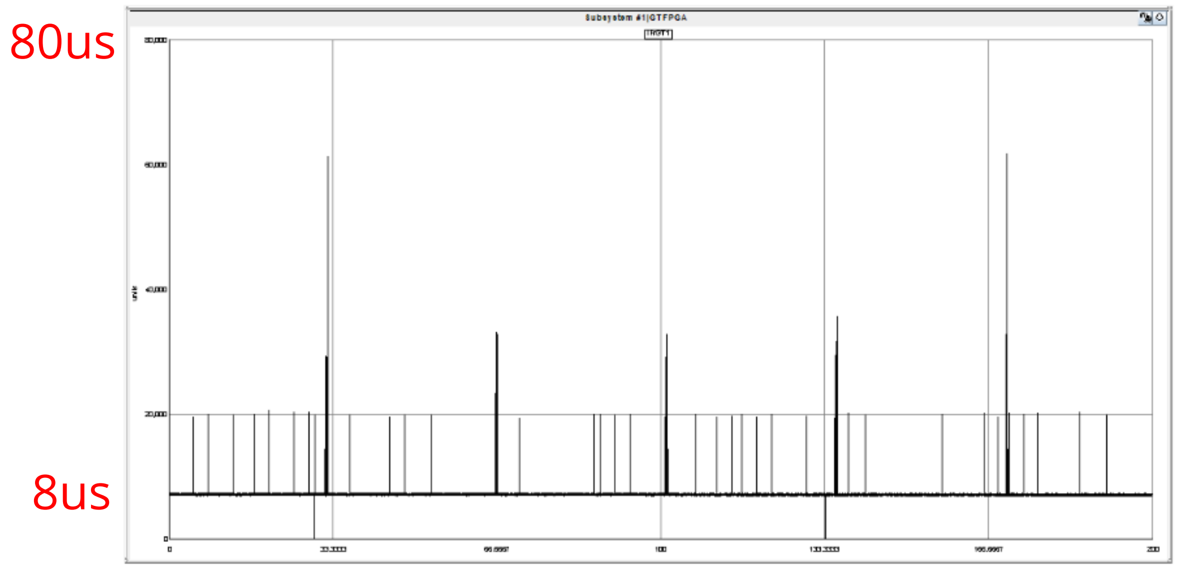
Black Box Controls with the RTDS

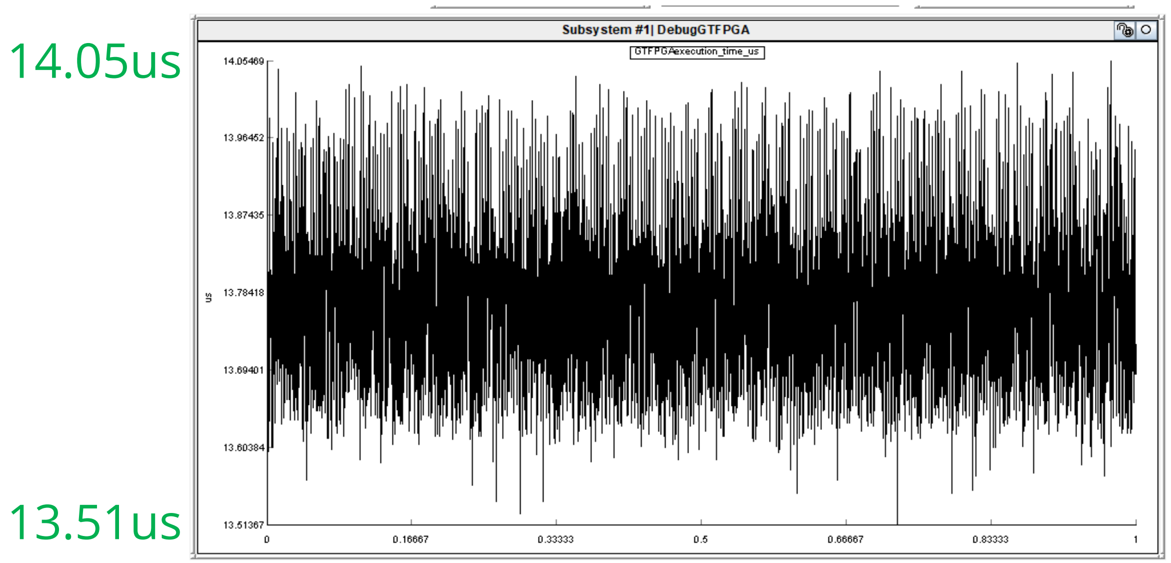
× **Linux OS running shared library (.so)**

The problem is the **indeterministic** execution time spike ~ 200 us, which is hard to eliminate without third-party real-time OS support.

✓ **Bare-Metal running static library (.a)**

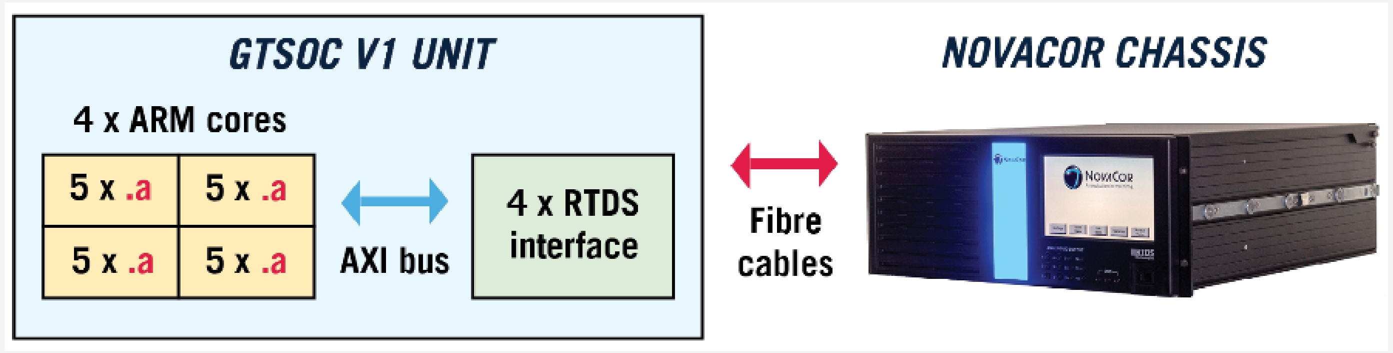
Bare metal guarantees **deterministic** timing: <1us spike.





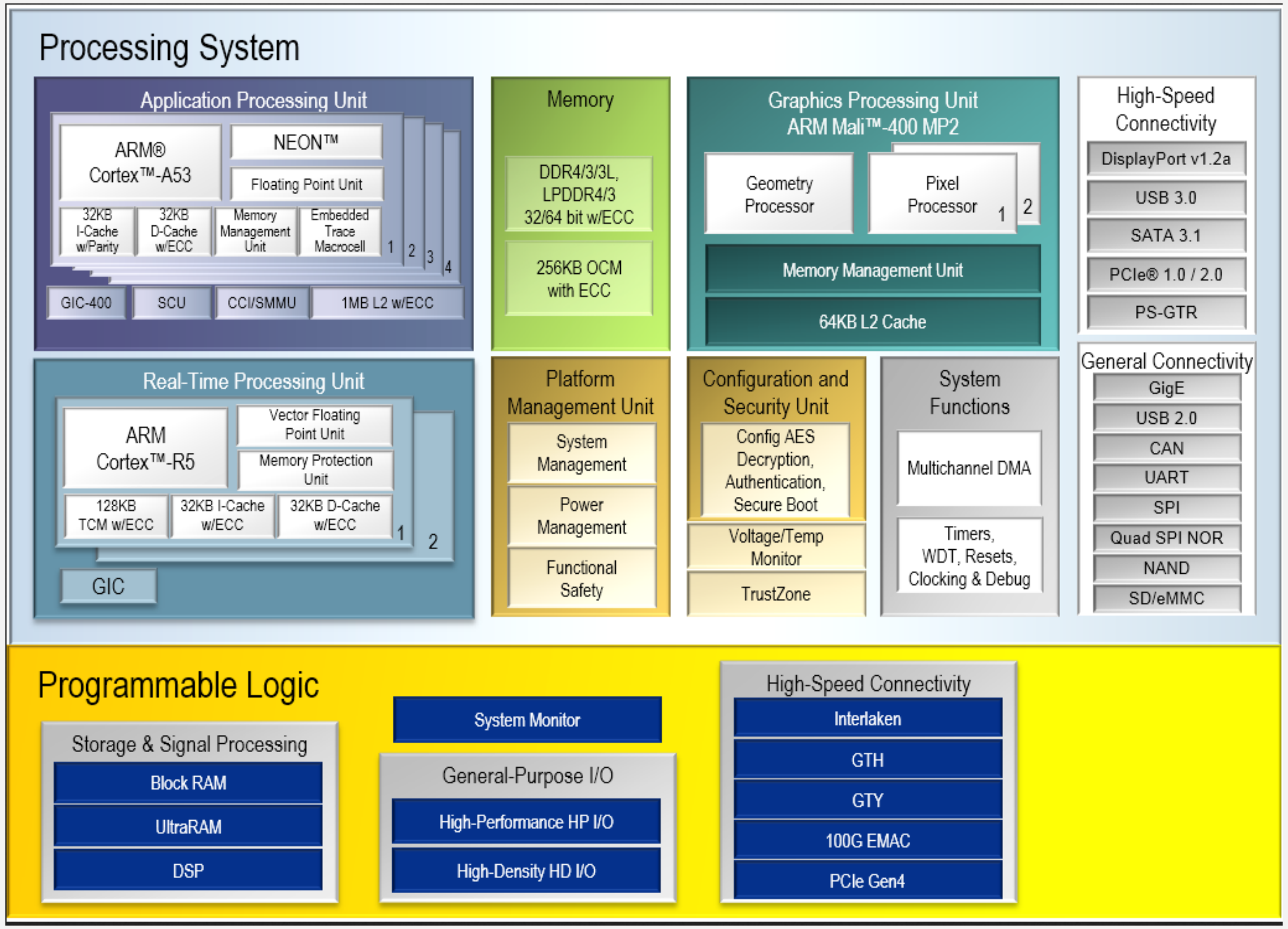
Introduction to GTSOC

- New hardware to support execution of black box controls
- GTSOC – System-on-Chip
- Combination of FPGA and Multi-Processor System-on-Chip (MPSoC)
- Supports Bare Metal Execution of Static Library (*.a) Files containing Vendor Source Code
- Interconnects with NovaCor via Fiber Optic Cables



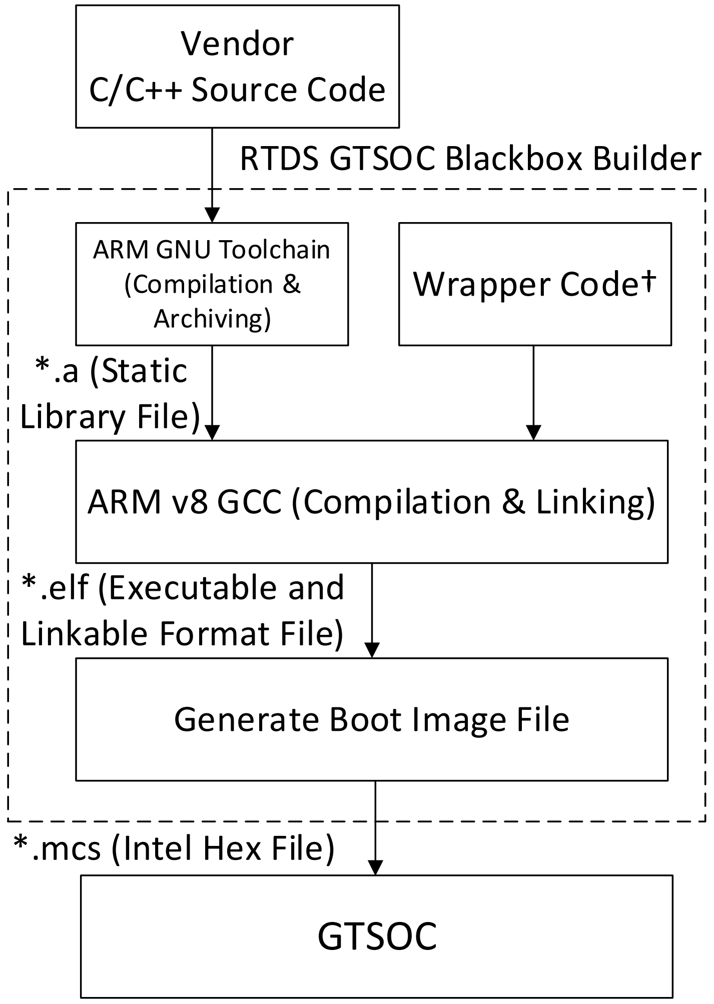
GTSOC Hardware

- **Xilinx Zynq UltraScale+ XCZU15EG**
- **Processing System (PS)**
  - 4 Application cores: ARM Cortex-A53
  - 2 Real-time cores: ARM Cortex–R5
  - High/low speed connectivity
- **Programmable Logic (PL)**



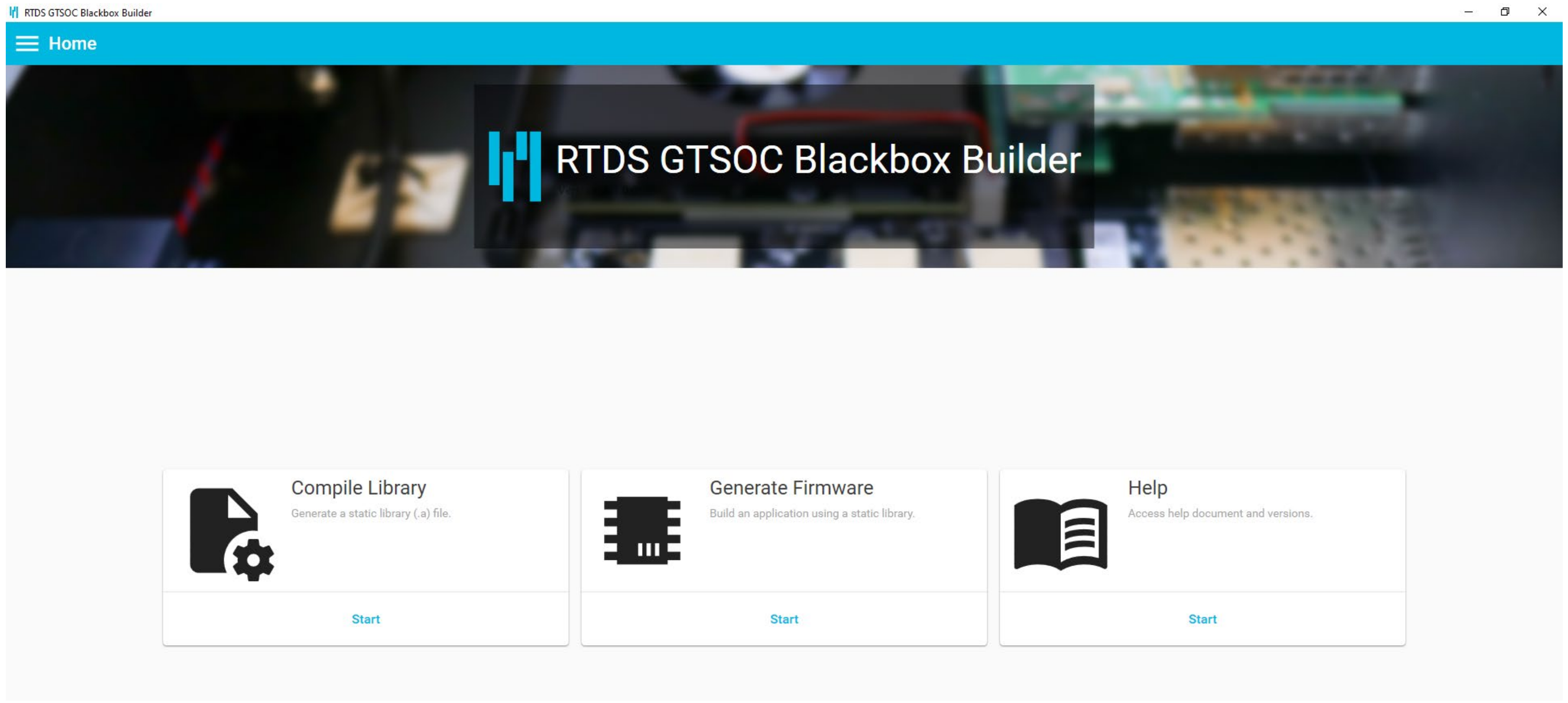
Deploying Blackbox Controls on the GTSOC

- Vendor C/C++ Source Code
  - Written by Hand
  - Generated via MATLAB/Simulink
- RTDS GTSOC Blackbox Builder Tool
  - Source code compiled into Static Library (*.a)
  - Wrapper code is used for mapping inputs/outputs and parameters
  - Generates Firmware (*.mcs) for GTSOC
- Templates are Provided for Wrapper Code


Vendor C/C++ Source Code
RTDS GTSOC Blackbox Builder
ARM GNU Toolchain (Compilation & Archiving)
Wrapper Code†
*.a (Static Library File)
ARM v8 GCC (Compilation & Linking)
*.elf (Executable and Linkable Format File)
Generate Boot Image File
*.mcs (Intel Hex File)
GTSOC

RTDS GTSOC Blackbox Builder

- Cross-compile C/C++ source code to static library (*.a) file
- Develop wrapper code
    - Templates provided
- Links the static library to build application executable .elf file
- Generate the GTSOC firmware (*.mcs) file
- Requires Xilinix Vitis Software

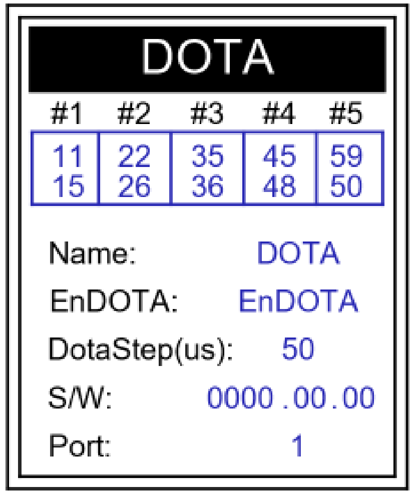


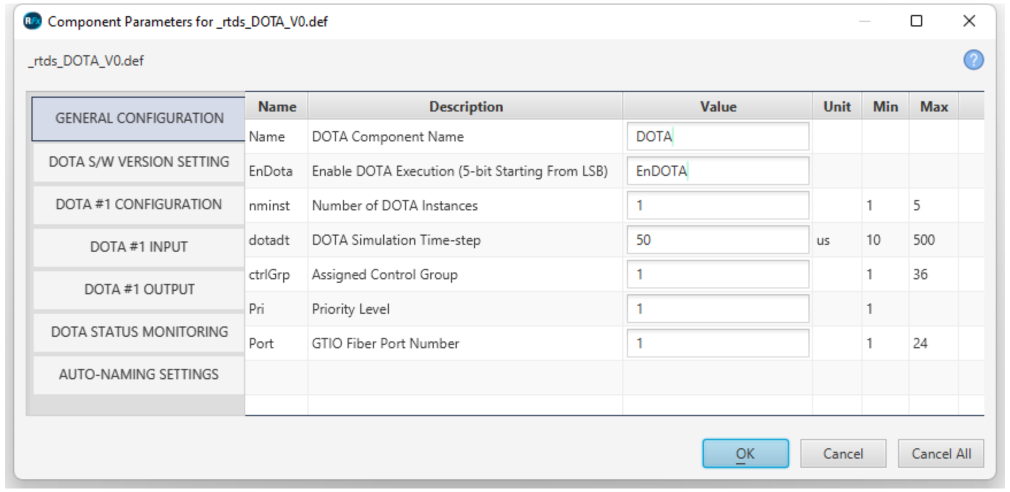
RTDS GTSOC Blackbox Builder
Home
RTDS GTSOC Blackbox Builder
Compile Library
Generate a static library (.a) file.
Start
Generate Firmware
Build an application using a static library.
Start
Help
Access help document and versions.
Start

RSCAD FX GTSOC DOTA Component

- Each GTSOC unit has one ARM Cortex-A53 Processsor with 4 cores
  - Each ARM Cortex-A53 core requires one DOTA component
    - Each DOTA component supports up to 5 DOTA instances
      - Each DOTA instance supports up to 64 inputs and 64 outputs
        - Each instance could be the same (multi-instance) or different
- Black Box Controller parameters can be read in via text file and used during initialization stage prior to execution

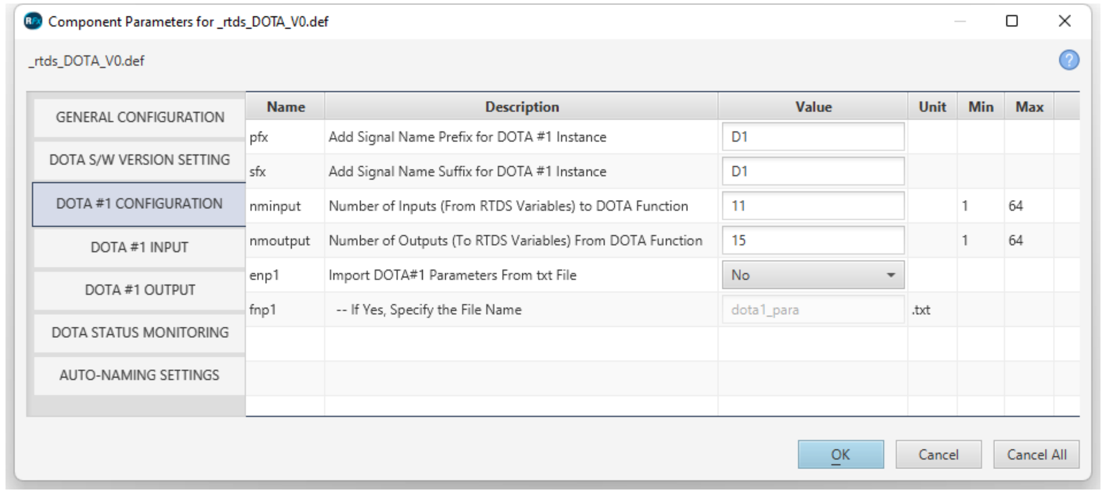
DOTA Component





Name	Description	Value	Unit	Min	Max
Name	DOTA Component Name	DOTA			
EnDota	Enable DOTA Execution (5-bit Starting From LSB)	EnDOTA			
nminst	Number of DOTA Instances	1		1	5
dotadt	DOTA Simulation Time-step	50	us	10	500
ctrlGrp	Assigned Control Group	1		1	36
Pri	Priority Level	1		1	
Port	GTIO Fiber Port Number	1		1	24

General Configuration

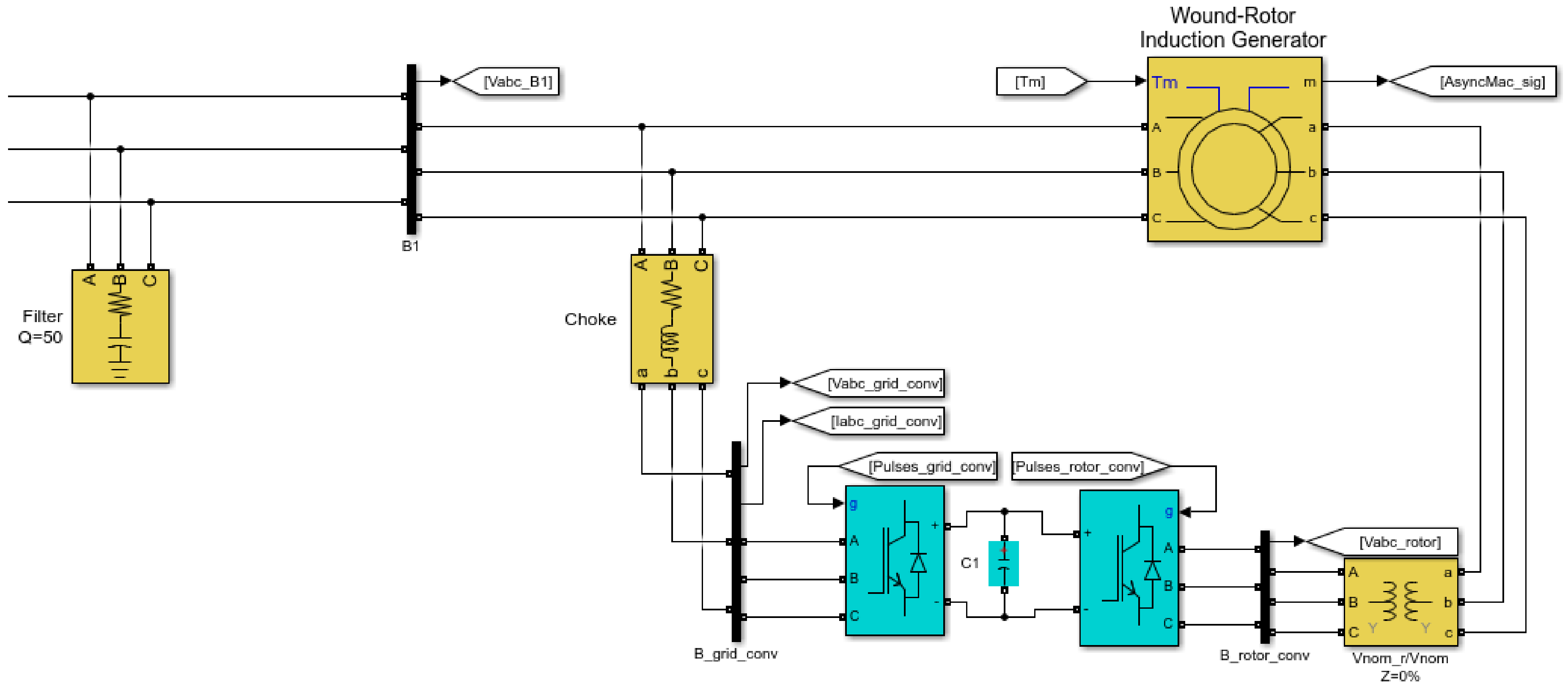


Name	Description	Value	Unit	Min	Max
pfx	Add Signal Name Prefix for DOTA #1 Instance	D1			
sfx	Add Signal Name Suffix for DOTA #1 Instance	D1			
nminput	Number of Inputs (From RTDS Variables) to DOTA Function	11		1	64
nmoutput	Number of Outputs (To RTDS Variables) From DOTA Function	15		1	64
enp1	Import DOTA#1 Parameters From txt File	No			
fnp1	-- If Yes, Specify the File Name	dota1_para	.txt		

DOTA Instance Configuration

DFIG System Example

Simulink DFIG System (Electrical System)

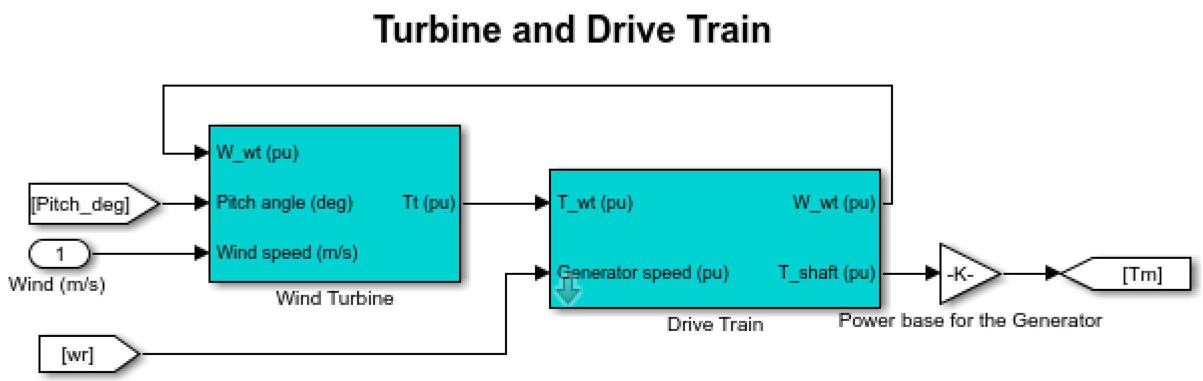


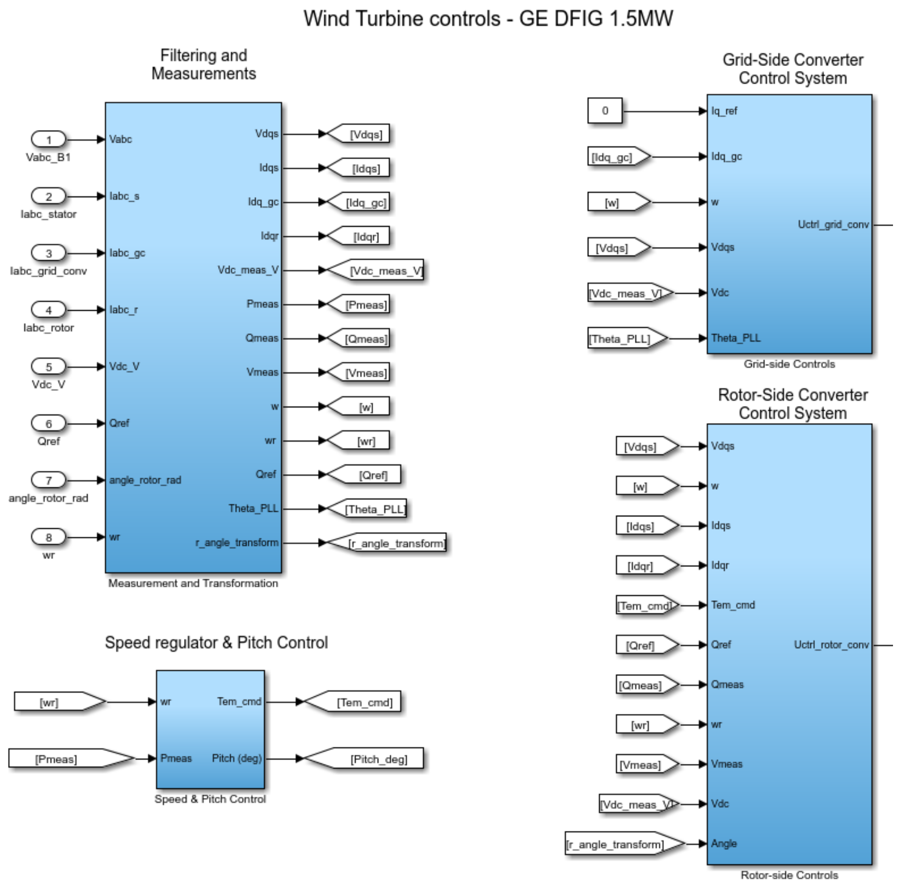
DFIG System Example

Simulink DFIG System (Control System)

Turbine and Drive Train:
- Wind turbine
- Drive train

Wind Turbine Controls
- Filtering and measurements
- Grid-side converter control system
- Rotor-side converter control system
- Speed regulator & pitch control





DFIG System Example

RSCAD FX DFIG System

Electrical System (on NovaCor):

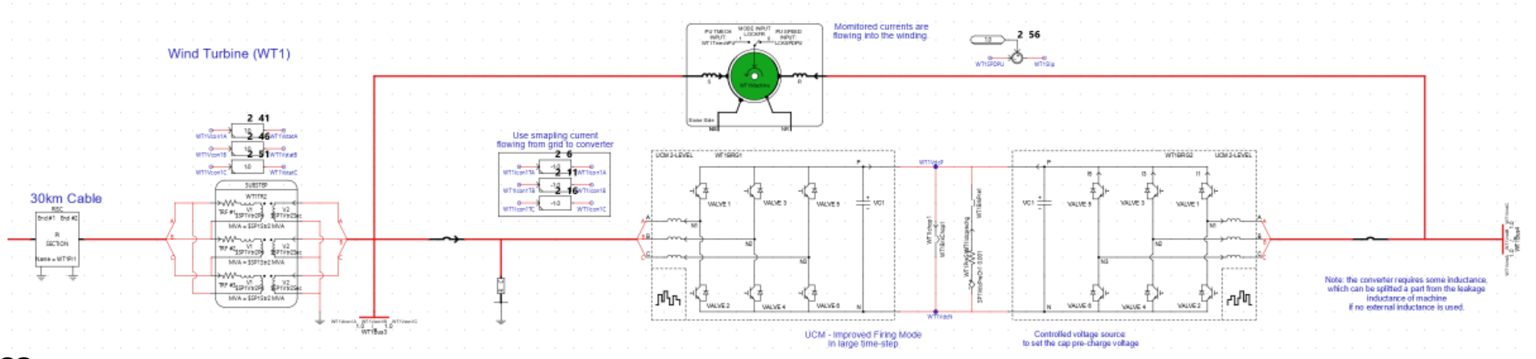
- Same as the circuit in Simulink

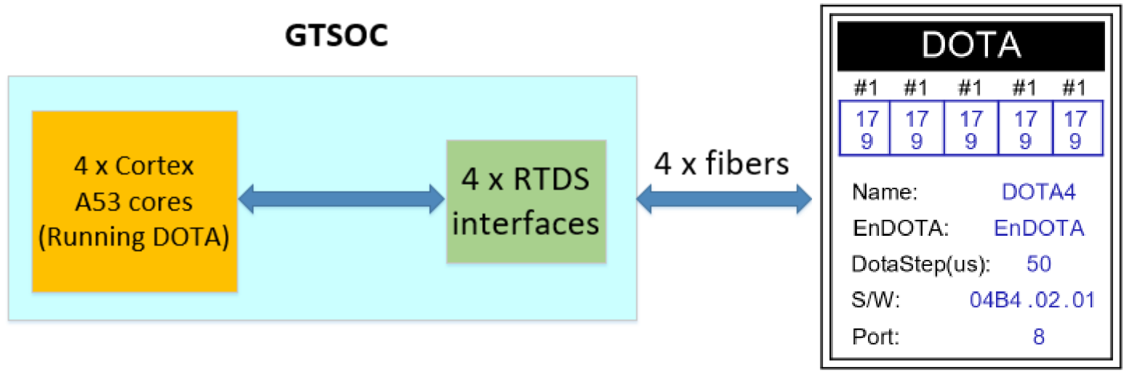
Control System (on GTSOC):

- Filtering and measurements
- Grid-side converter control system
- Rotor-side converter control system
- Speed regulator & pitch control

Interface (NovaCor & GTSOC)

- DOTA component: RTDS interface (Port 1-20)


Wind Turbine (WT1)
30km Cable
Use smapling current flowing from grid to converter
Monitored currents are flowing into the winding.
UCM - Improved Firing Mode in large time-step
Controlled voltage source to set the cap pre-charge voltage
Note: the converter requires some inductance, which can be splitted a part from the leakage inductance of machine if no external inductance is used.



Name:	DOTA4
EnDOTA:	EnDOTA
DotaStep(us):	50
S/W:	04B4 .02.01
Port:	8

DFIG System Example

Multi-Core and Multi-Instance Testing

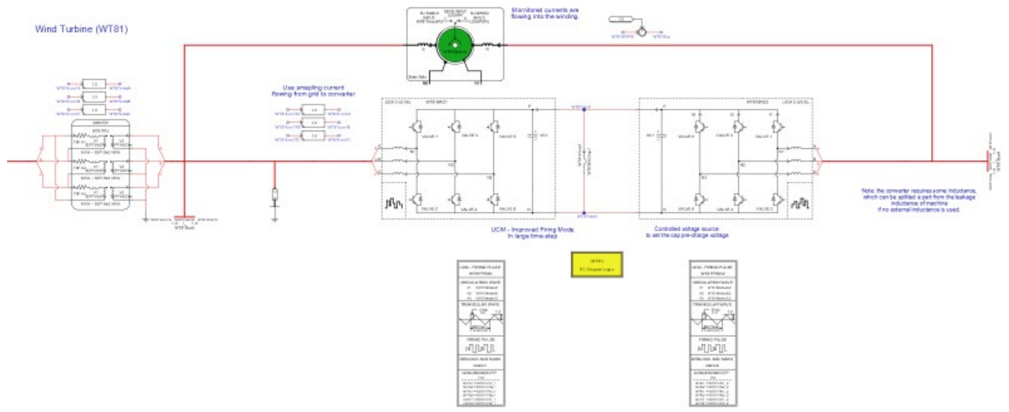
Electrical system :

- 21 DFIG systems, 2 AC Thevenin networks

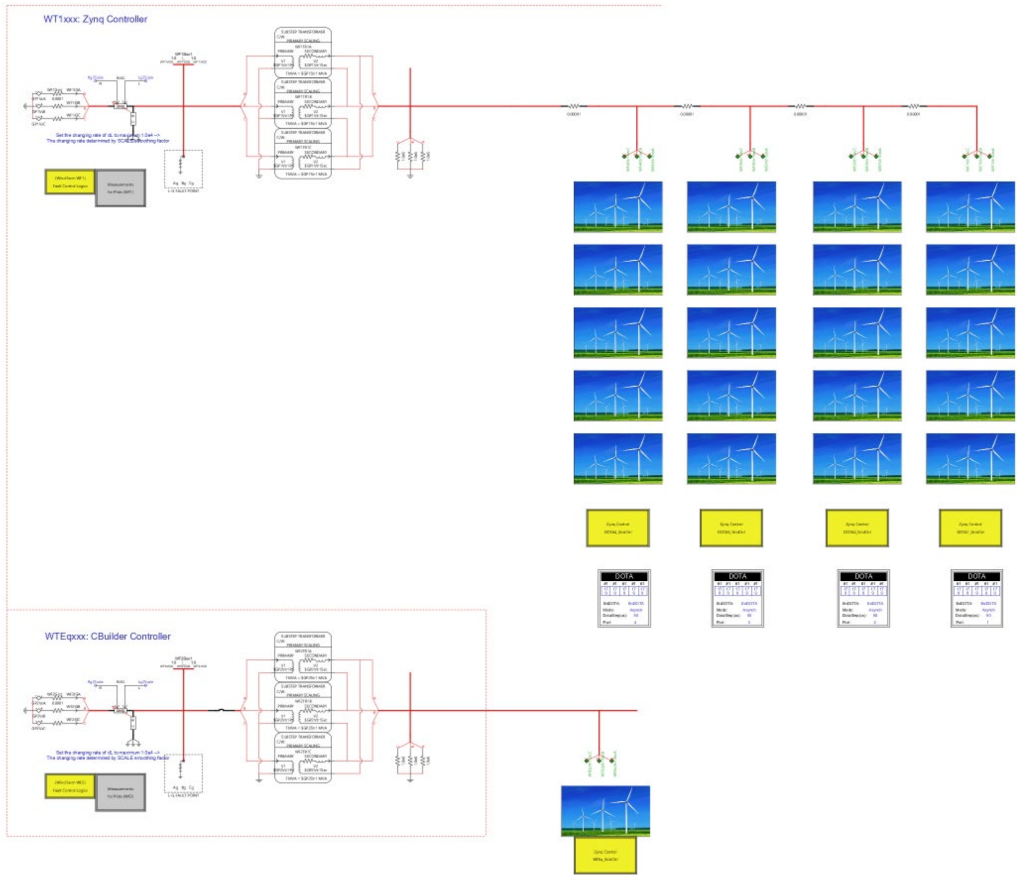
Control system:

- 20 DFIG controls on 4 GTSOC cores (every 5 DFIGs controls on one GTSOC core)

Four DOTA components (one per GTSOC core, each has 93 outputs and 53 inputs**)**



Single DFIG Electrical System



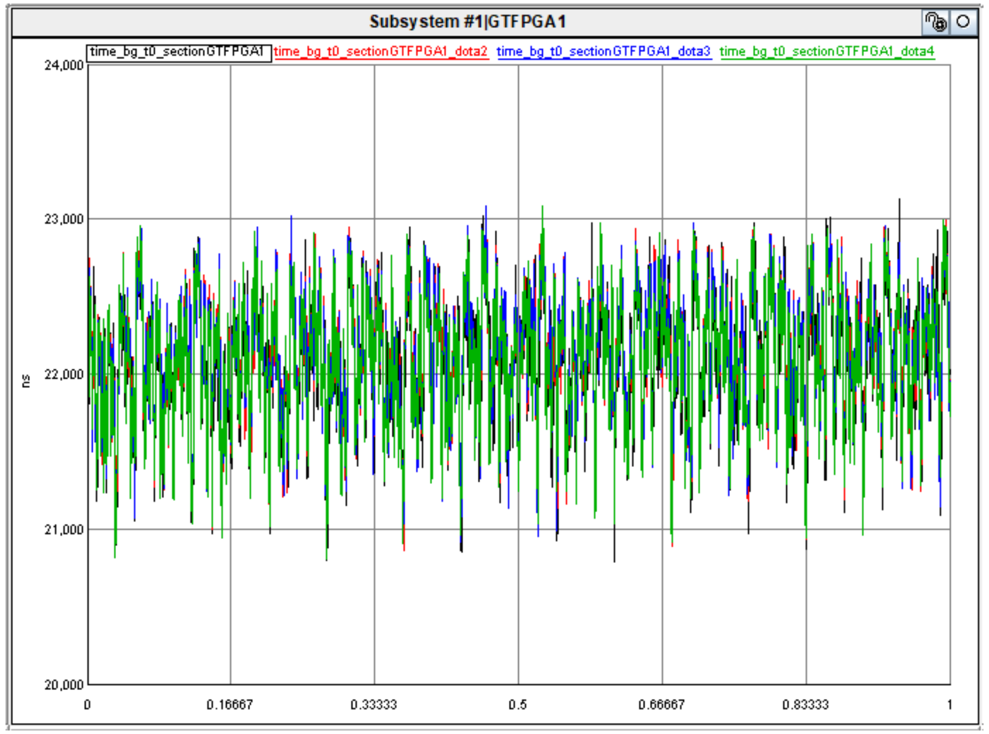
21 DFIG Electrical System

DFIG System Example

Multi-Core and Multi-Instance Testing - Timing

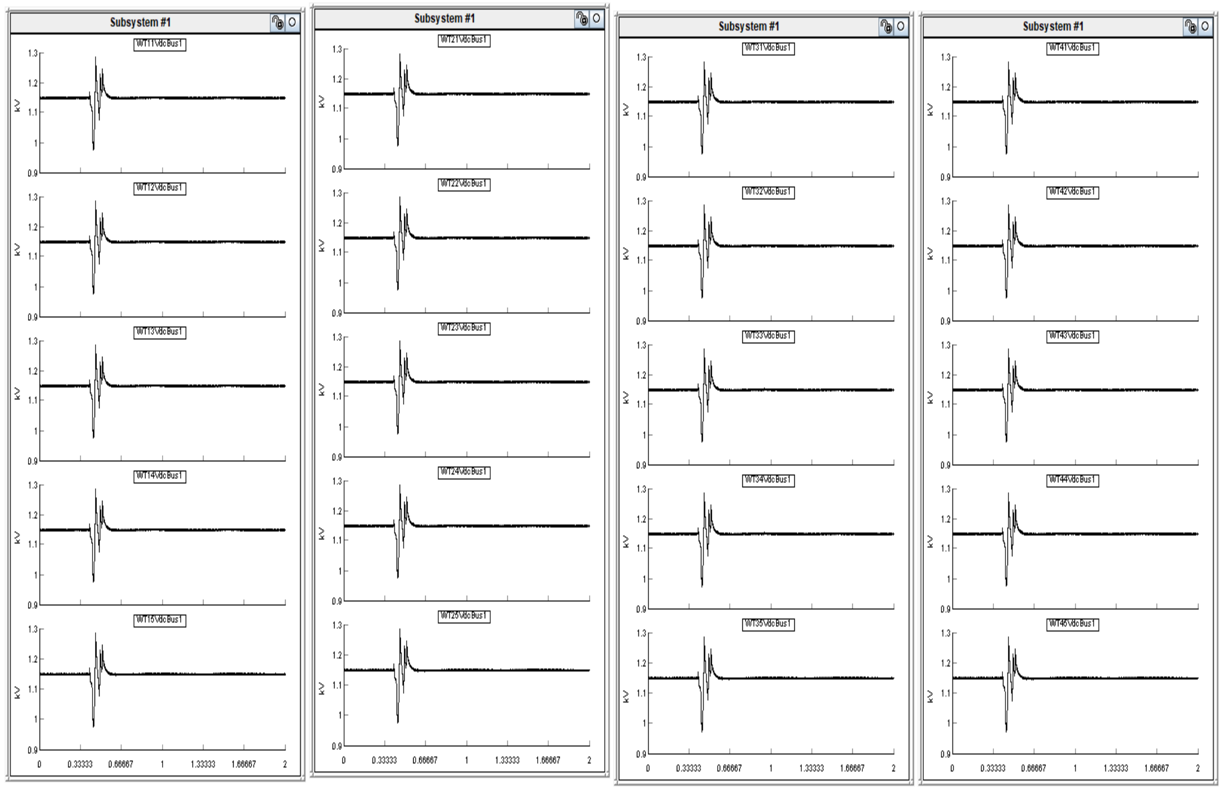
20 DFIG DOTA Timing + Communication Timing (93 Outputs +53 Inputs): 22us ± 1us jitter

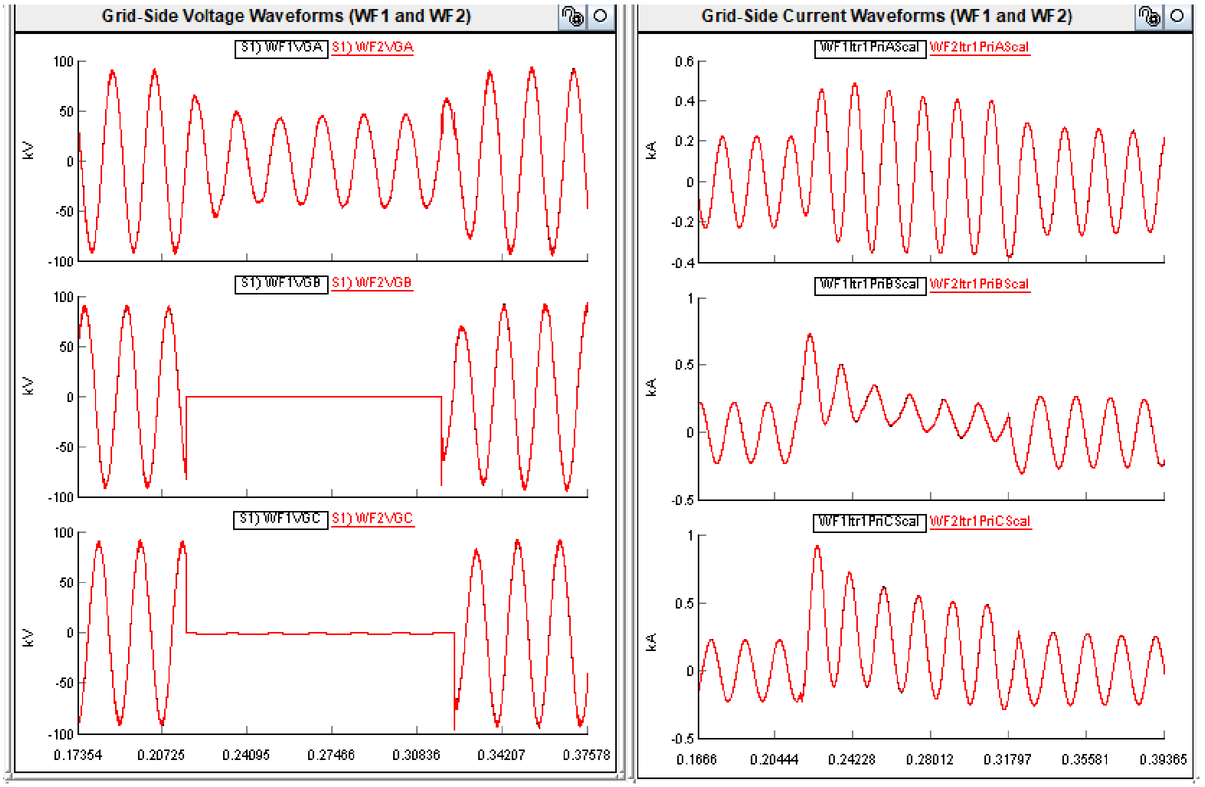
- Communication : 6.3us
- Each DFIG DOTA : (23-6.3)/5 = **3.34us**



DFIG System Example

Multi-Core and Multi-Instance Testing – AC Fault

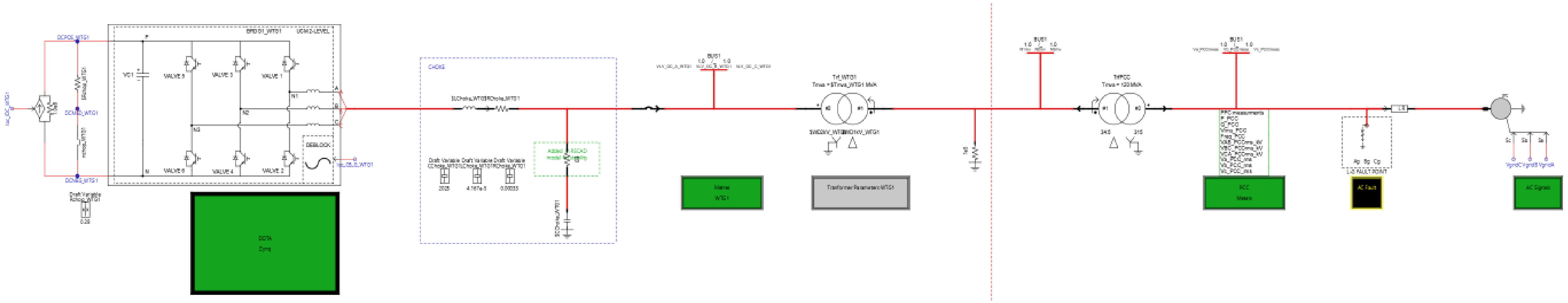




Fault on the AC Thevenin **System 1 and 2** (20 DOTAs on 4 GTSOC cores)

Wind Turbine System Example

RSCAD FX Case



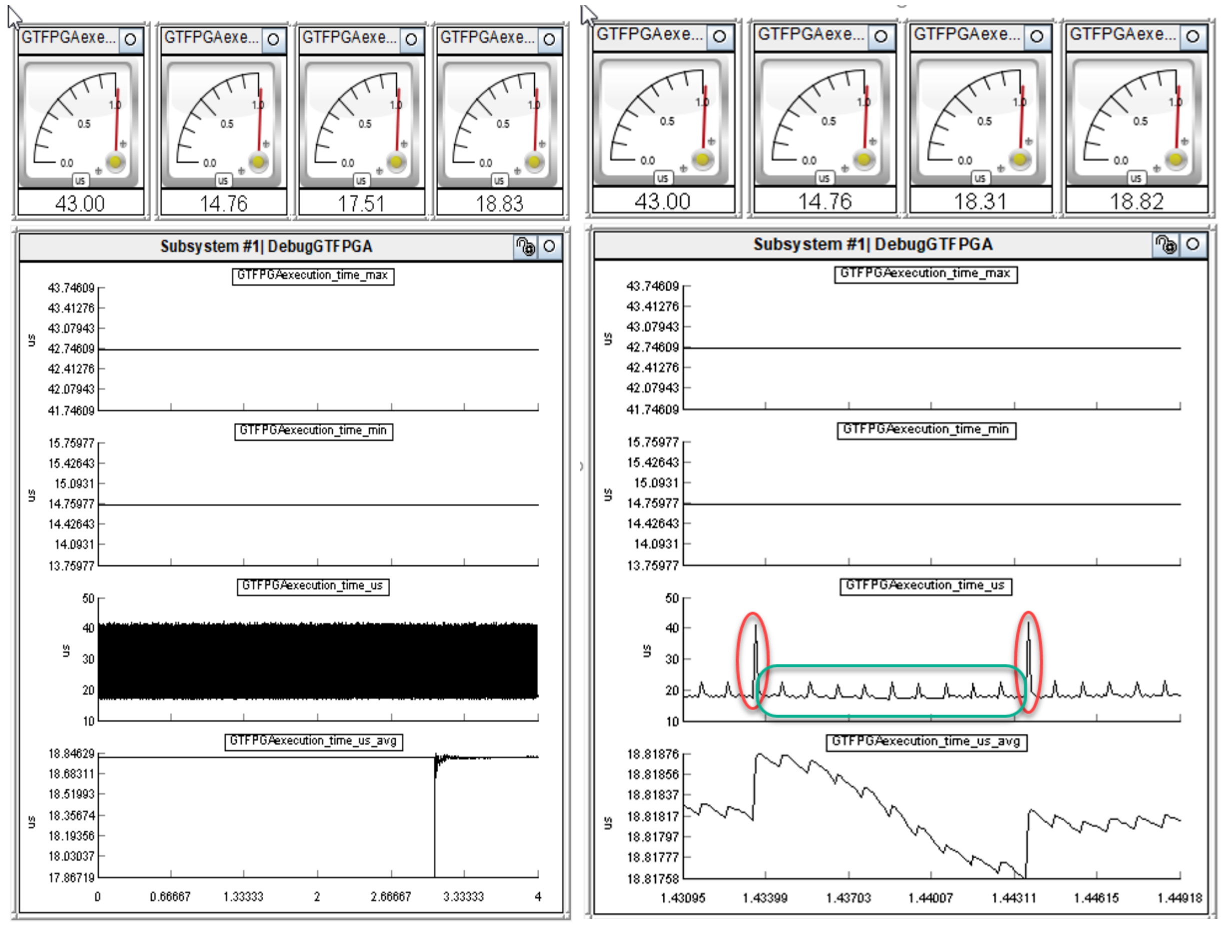
For the wind turbine energy system:

- **Current injection** to represent the machine-side converter
- **UCM** to represent the grid-side converter
- Grid-side transformer, windfarm Integration transformer, and Thevenin equivalent AC source

- GTSOC (DOTA) is used for the UCM converter control.
- Simulation time step size: 100us

Wind Turbine System Example

Timing Statistics



DOTA execution time has a 10ms (red circle) and 1ms (green circle) periodic jump.
This is not spike or jitter but Customer's scheduled tasking.

Max execution time: 42.7us

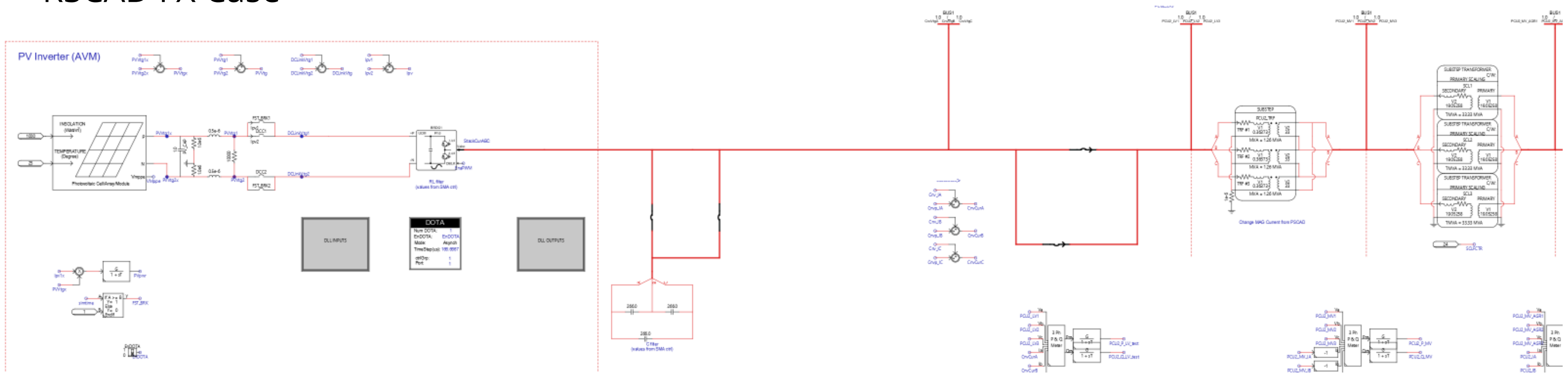
Min execution time: 14.8us

Instantaneous execution time: 18us

Average execution time: 19us

SMA Inverter Example

RSCAD FX Case

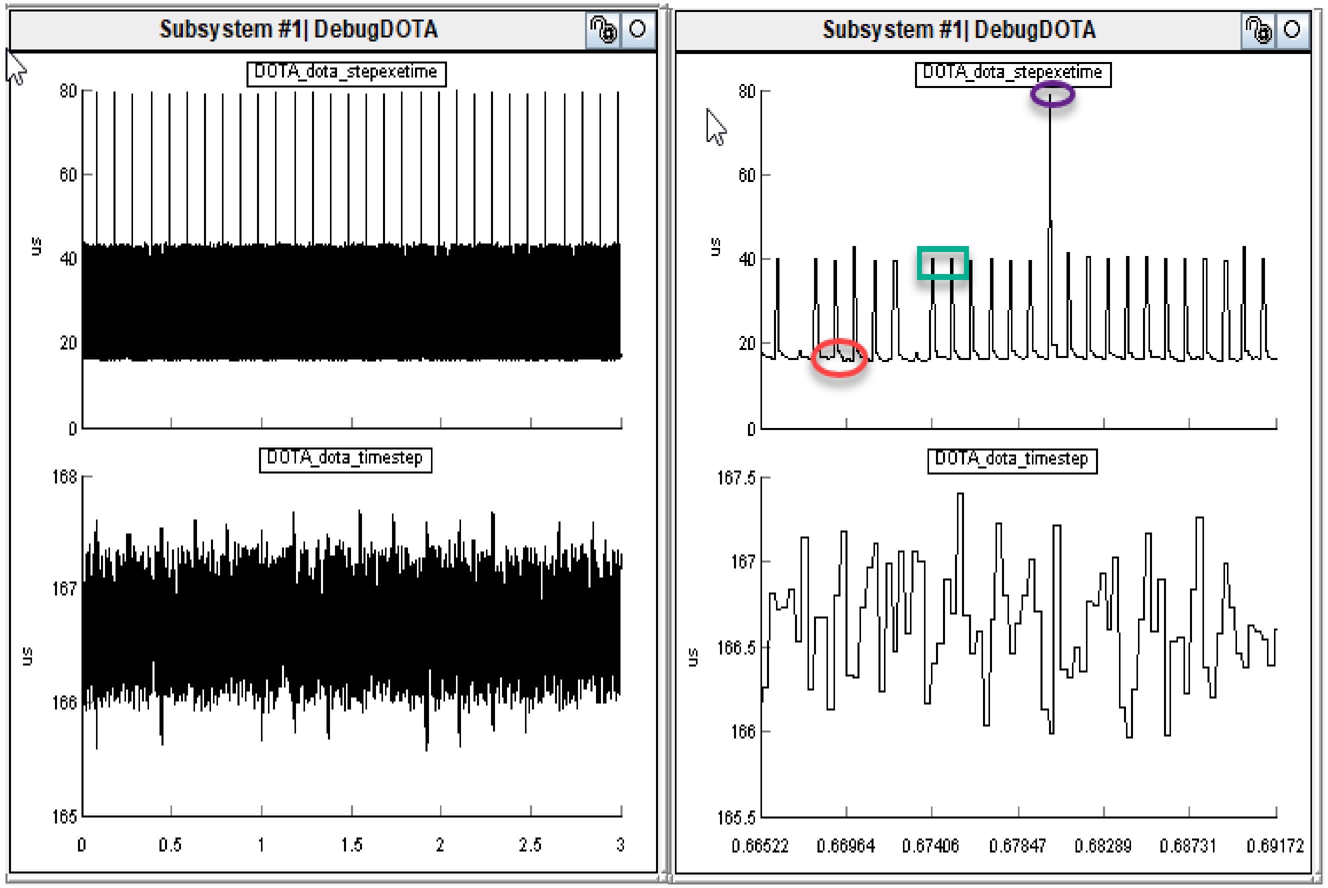

PV Inverter (AVM)
INSOLATION
TEMPERATURE
Photovoltaic Cell/Array/Module
DOTA
DLL INPUTS
DLL OUTPUTS
RL filter (values from SMA ckt)
C filter (values from SMA ckt)
SUBSTEP TRANSFORMER
Change MAG Current from PSCAD

For the PV system:

- PV panel
- UCM to represent the grid-side converter
- Grid-side transformer, scaling transformer, and Thevenin equivalent AC source
- GTSOC (dota) time step size: 166.67us.
- Simulation time step size: 50us

SMA Inverter Example

Timing Statistics



DOTA execution time: periodic jumps (purple and green) are not spikes but scheduled controller tasking (e.g., protection).

THANK YOU!
QUESTIONS?