

# STAPL++ Player Strategy

Ideas on Program Design and Data Structures

# Why STAPL?





- The embedded test is:
  - Board unique test vectors/program
  - Generic SW for vector delivery (Player)
- STAPL is de-facto, widely used standard (mainly for programming)
- Freeware:
  - ASCII and binary players
  - SVF to STAPL conversion (all commercial BScan ATPG can generate SVF, so interconnect test would be covered)
- STAPL can represent:
  - Vectors (like SVF)
  - Variables
  - Flow control

# BScan Vectors and Test Control SW

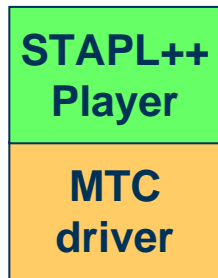
- Vectors/Test program language based on STAPL (Standard Test and Programming Language) = vectors + flow control
- Extensions to manage instantiation of components and DFT IP, and parallel procedures for DFT IP
- Allows reuse of procedures, and simplifies test programming
- Allows interactive operation on individual DFT IP in production test diagnosis and at I&V lab
- Unique STAPL++ program for each board
- Common STAPL++ player for all boards

# MSc Experiment

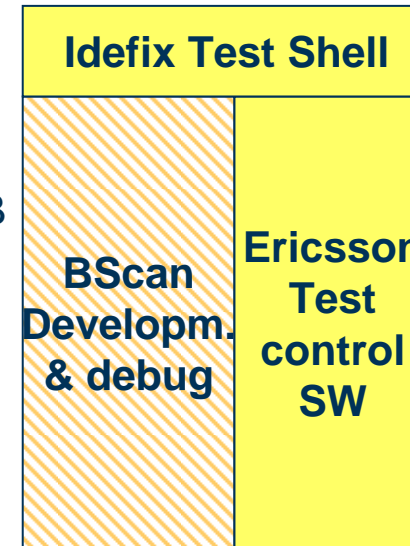
## Legend

-  = Test Platform
-  = Board SW
-  = Modified TP SW
-  = New SW (Altera based)

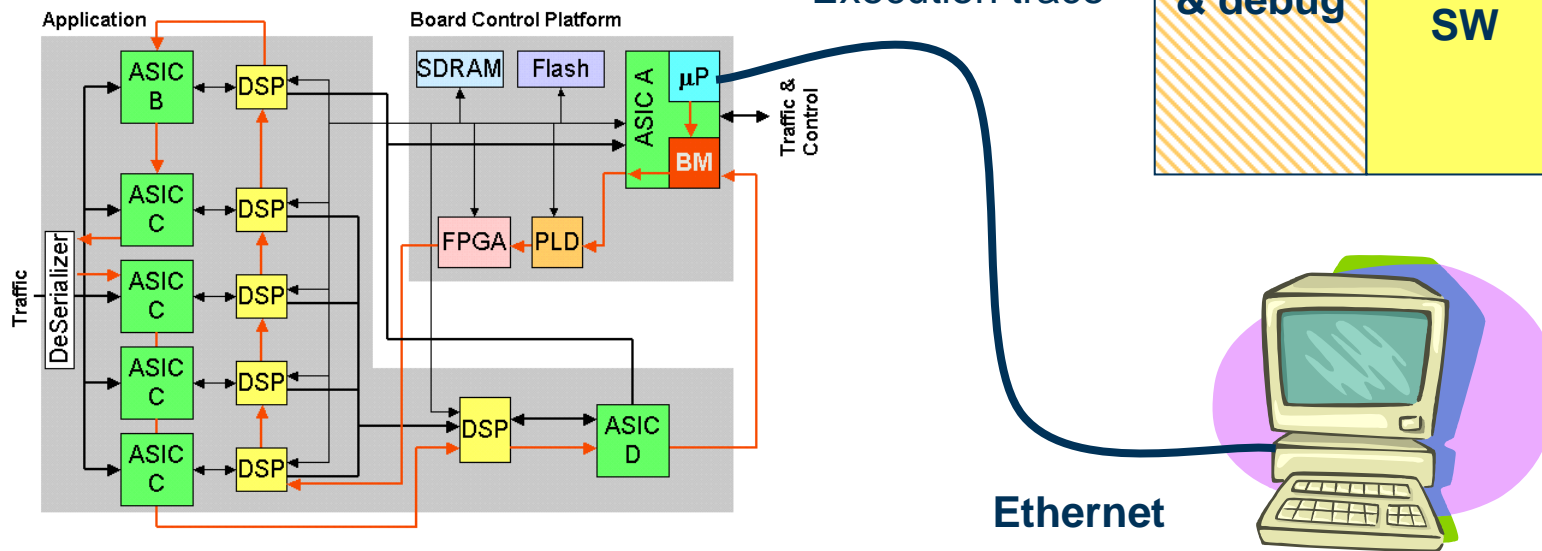
## PPC based SW "Test Controller"



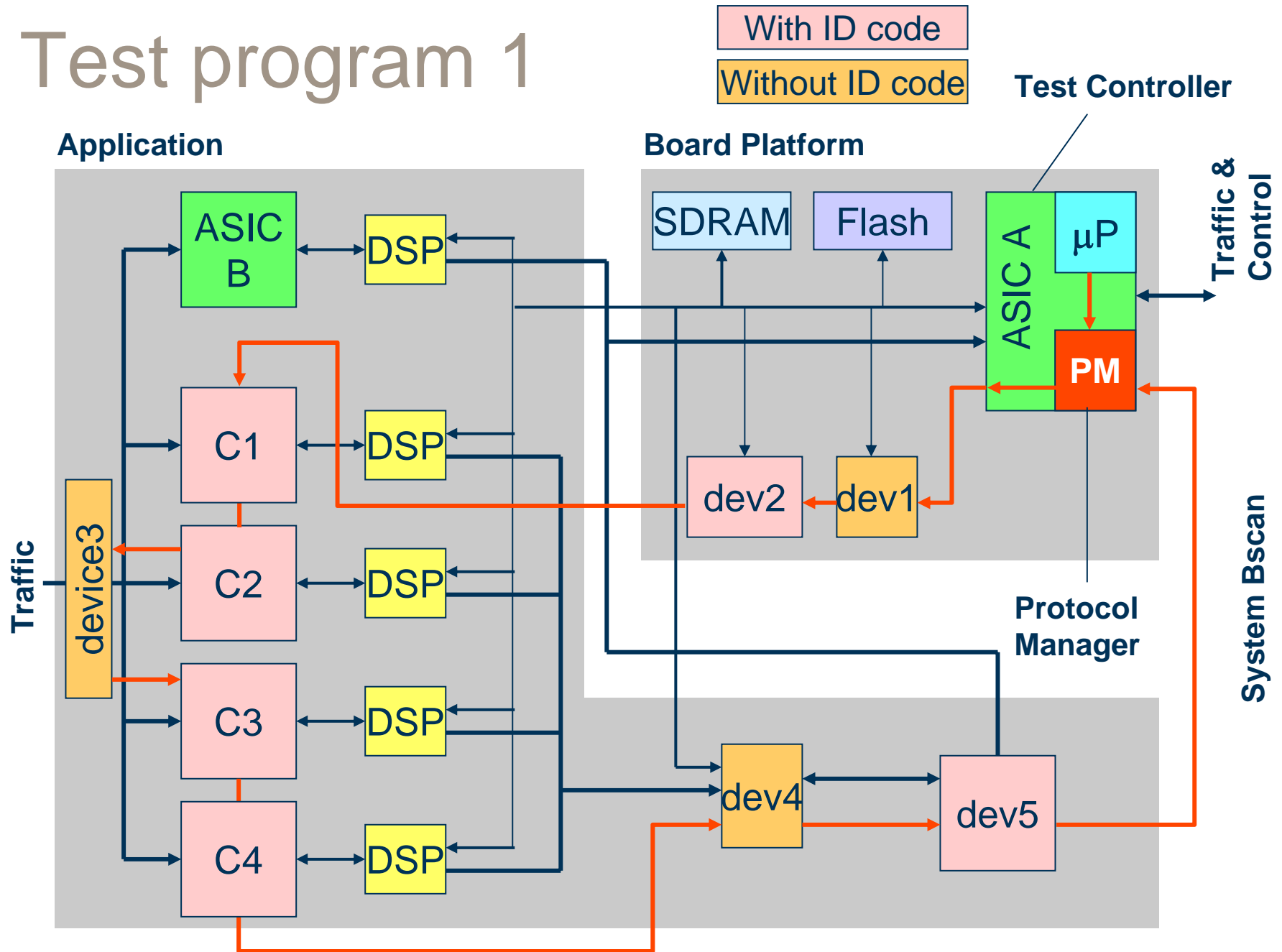
## PC based SW "Test Manager"



Commands:  
 Stop-on-fail  
 Run Action A  
 Run Procedure B  
 Return:  
 Failing vector  
 Execution trace

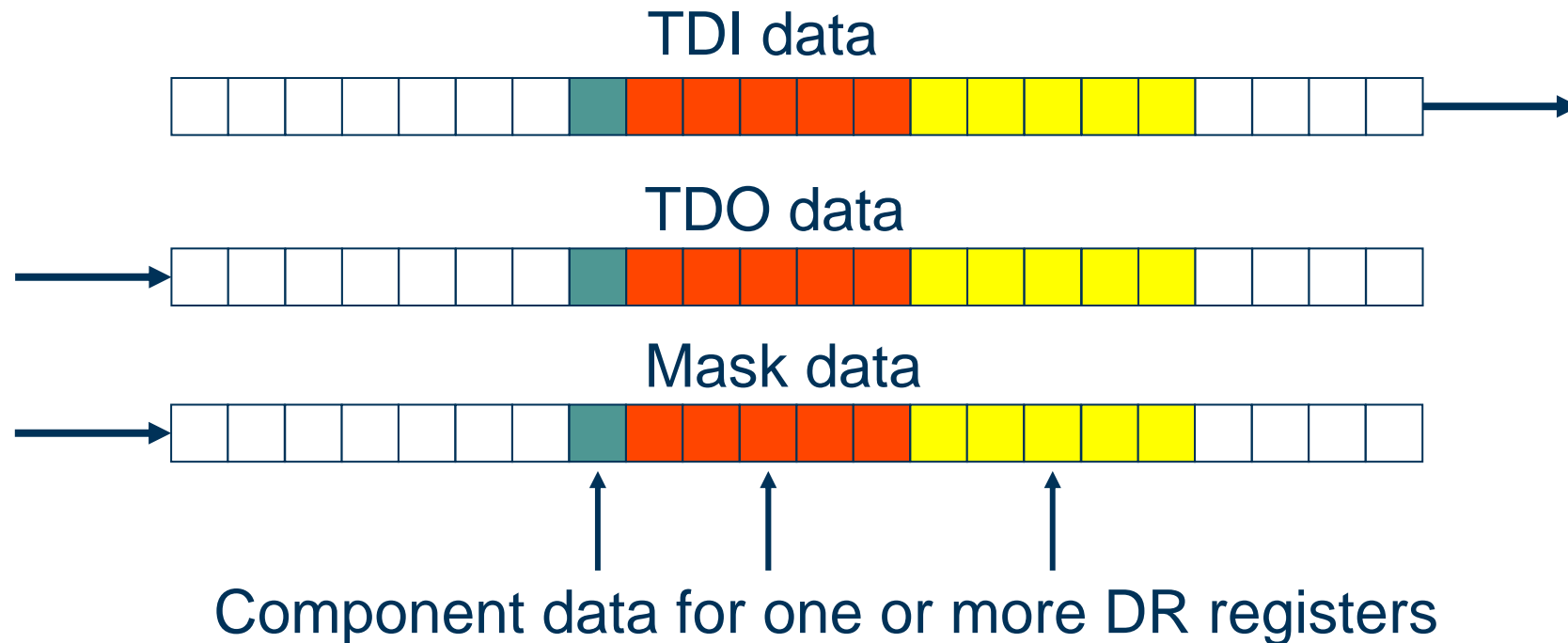


# Test program 1



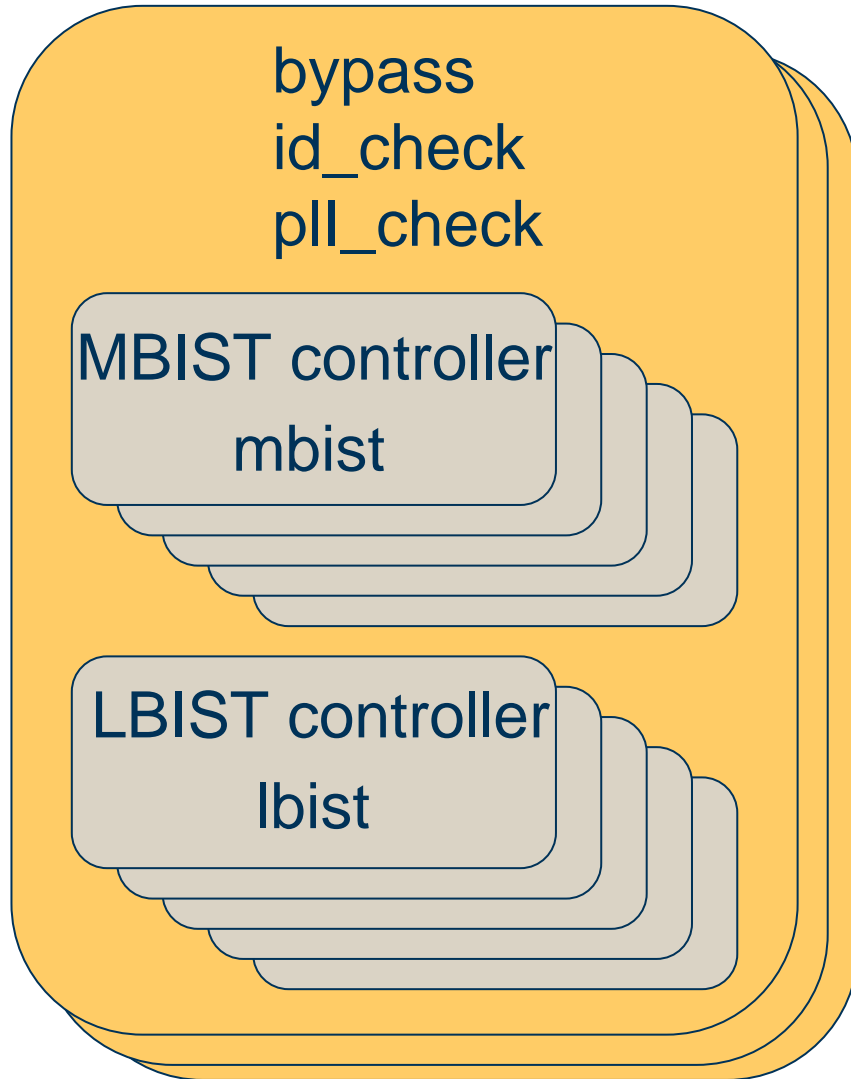
# Mapping component data to the Scan Chain – Typical STAPL DRSCAN statement

”Manual” composition of the scan data arrays –  
Knowledge of position in the overall chain is required

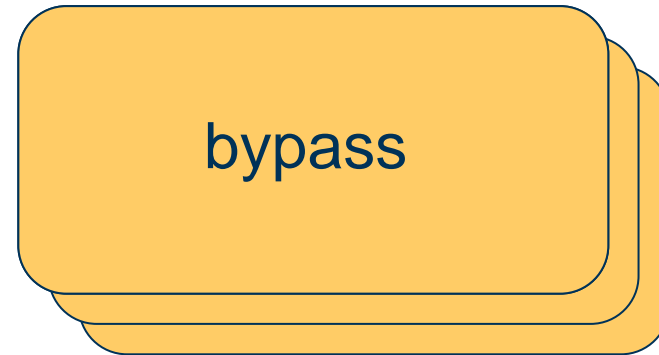


# Test Program, Components, DFT IP

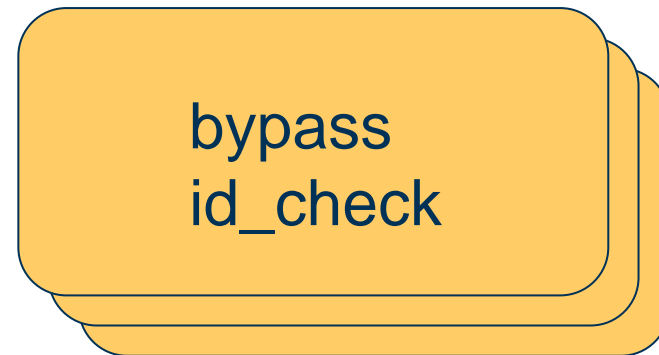
C



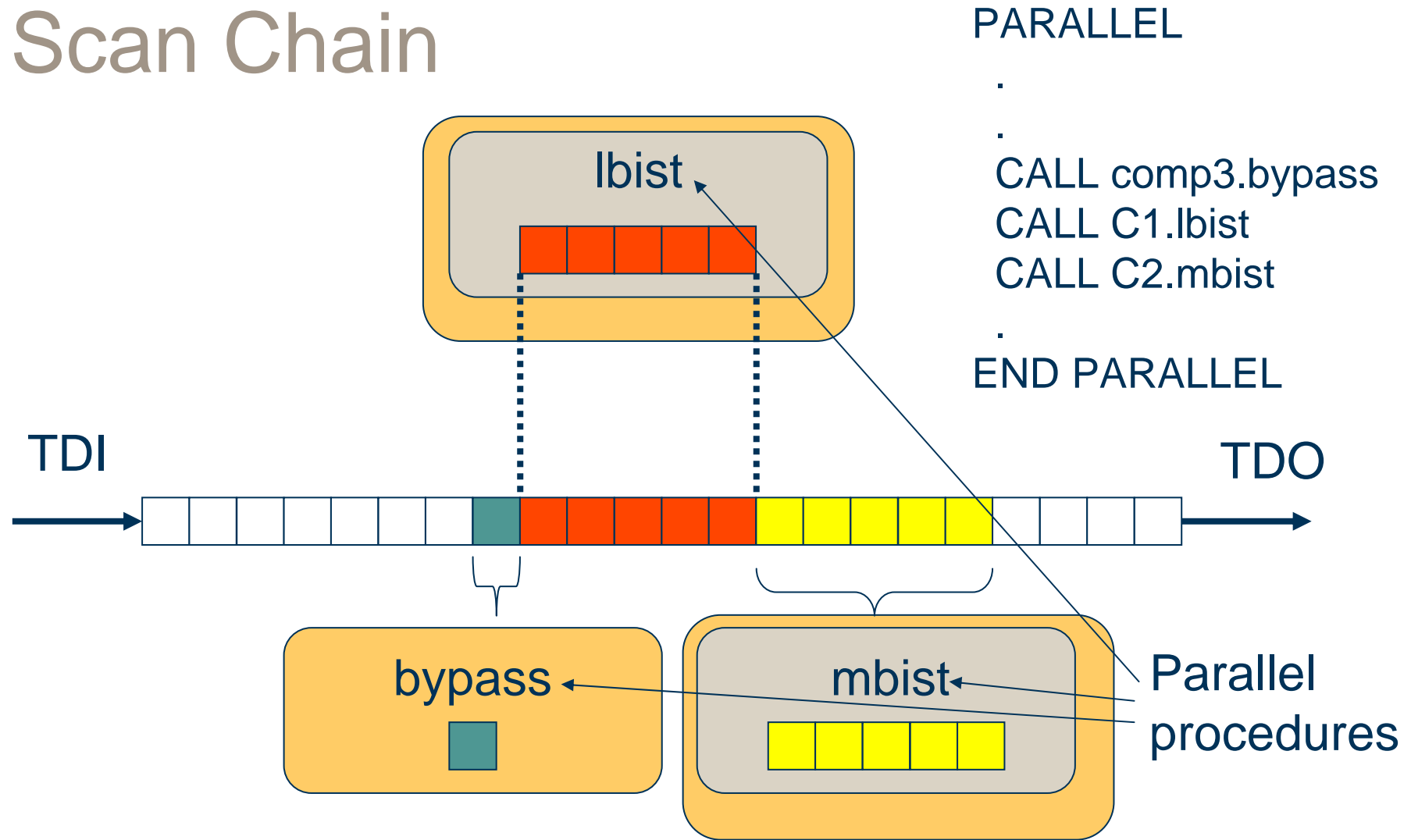
no\_id



has\_id

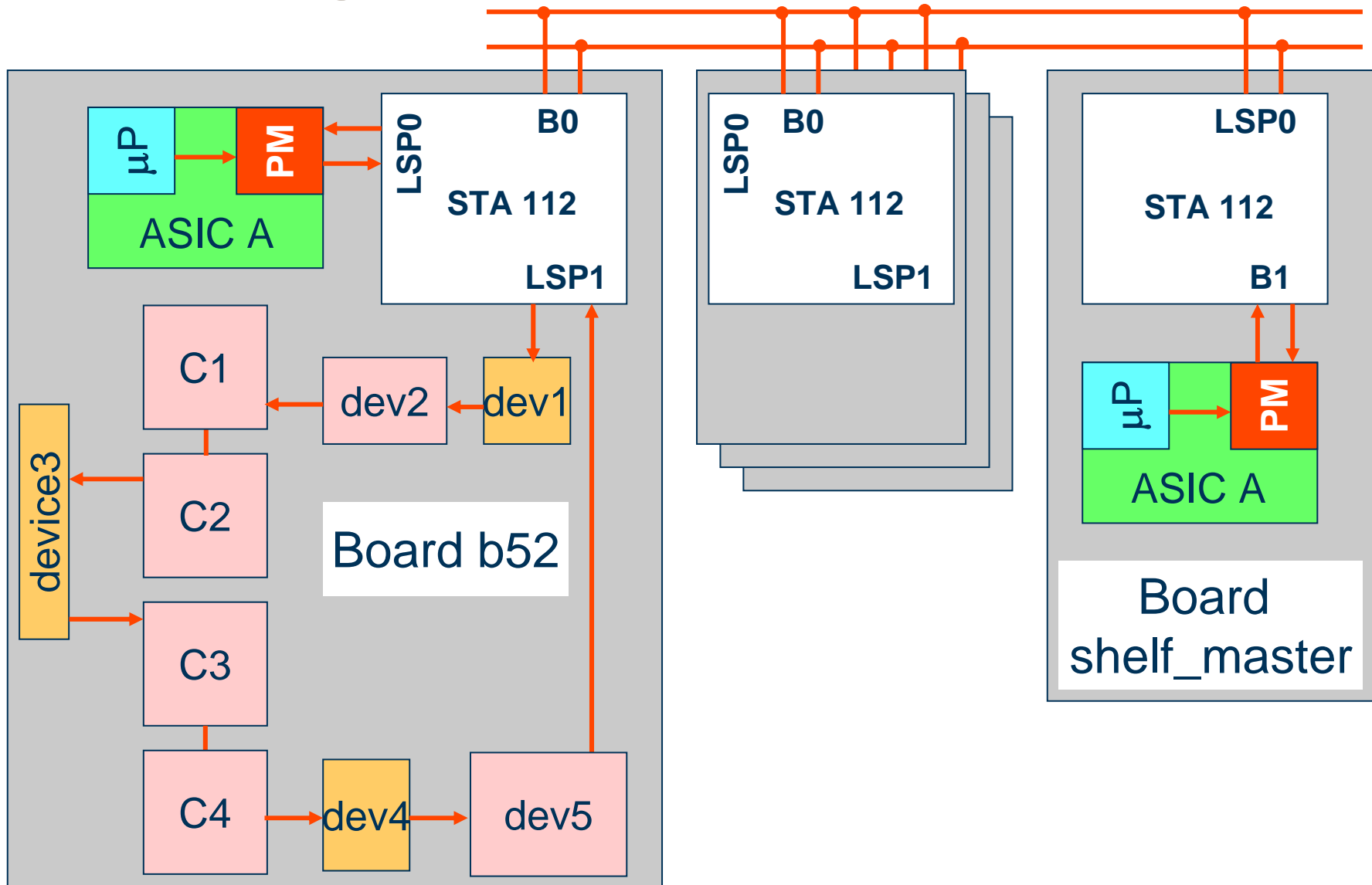


# Mapping procedures to the Scan Chain





# Test program 2



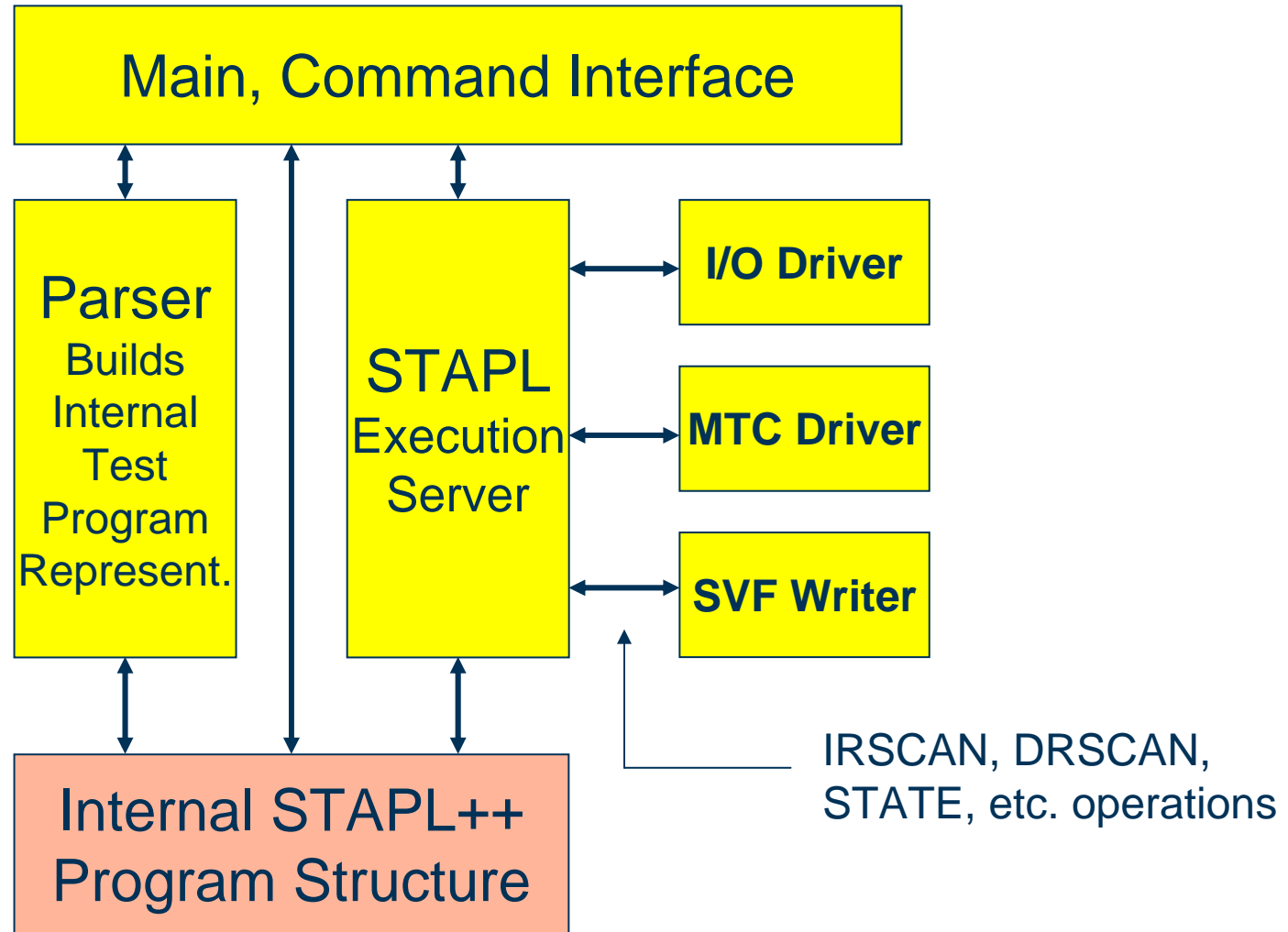
# Overall STAPL Structure

- NOTE statements
  - Describes file contents and features
- ACTION statements
  - Describes a sequence of steps of a complete operation
  - An ACTION is e.g. programming of a device (or several), test of interconnections, running BIST on a device
  - Is a list of PROCEDURE calls
- PROCEDURE blocks
  - Contains a flow of computations and actions on the BScan bus
  - May have flow control statements (e.g. IF, FOR)
  - Uses data from DATA blocks
  - May call other PROCEDURES
  - May use PRINT and EXPORT statements
- DATA blocks
  - Declarations of variables and constants
  - A compaction algorithm may be used
- CRC statement

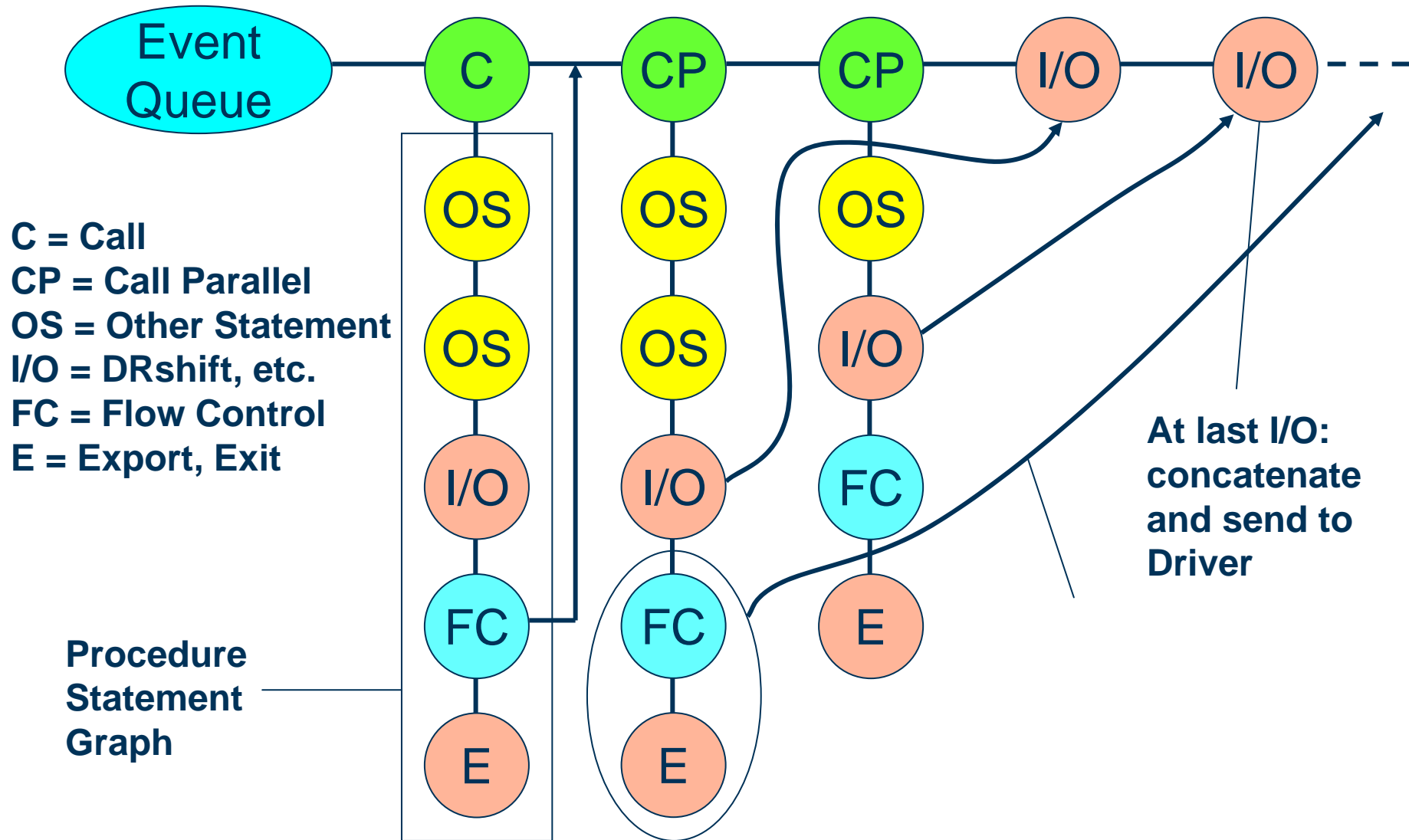
# Extended STAPL Features

- Structure Mapping
  - CLASS and INSTANTIATE maps components and hierarchical IC blocks to procedures and individual data
  - BODY executes at instantiation
  - component1.module3.lbist\_iterations = 1024
  - CALL component1.module3.lbist
- Parallel Execution and Synchronization
  - PARALLEL
    - CALL component1.bypass
    - CALL component2.bypass
    - CALL C1.lbist
  - IRSCAN and DRSCAN operations are concatenated and synchronous
- Backward Compatibility
  - Old STAPL programs will work with new player
  - Top CLASS and instance is implicit
- Compiled STAPL
  - Any STAPL program will compile to SDF and compact SDF-like formats, like EVF and BVF, unless they use TDO data for program flow control

# Next Step, Overall Structure



# Execution Flow and Parallelism



**ERICSSON** 

**TAKING YOU FORWARD**