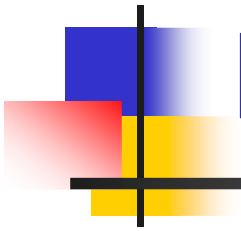


Building a Test Plan

Alon Shtepel

Yaniv Barash

Test Plan vs. Verification Plan

- Verification Plan (Environment Architecture):
  - Derived from System and Macro Spec.
  - High level description.
  - Targets, goals, feasibility
- Test Plan:
  - Detailed list
  - Derived from all documents.

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®

Test Plan Specifications

- Resources:
  - MRD, System Spec., Hardware Specs (Macro and Micro)
  - Designers notes.
  - Common tests for all chips.
  - Experience.

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®

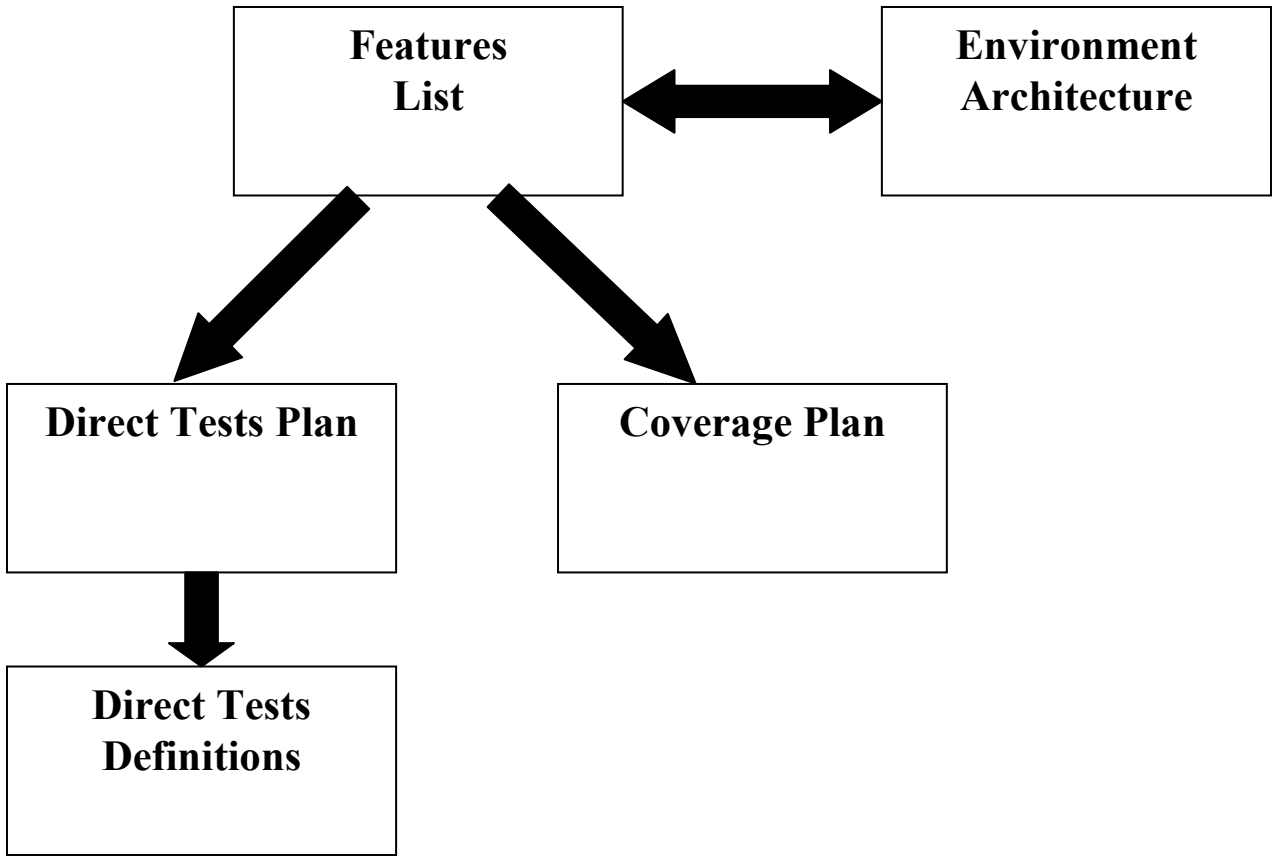
Test Plan Specifications (2)

- Consider:
  - IPs, Reused blocks, New blocks
  - Verification methodology - Random/direct
  - Granularity – System, Asic, Block, Unit.
  - Verification capabilities

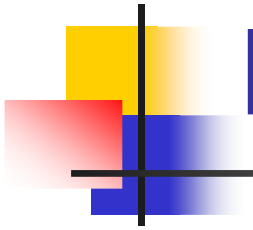
Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®

Documentation Flow



Discretix
embedding security solutions
MARVELL
MOVING FORWARD FASTER



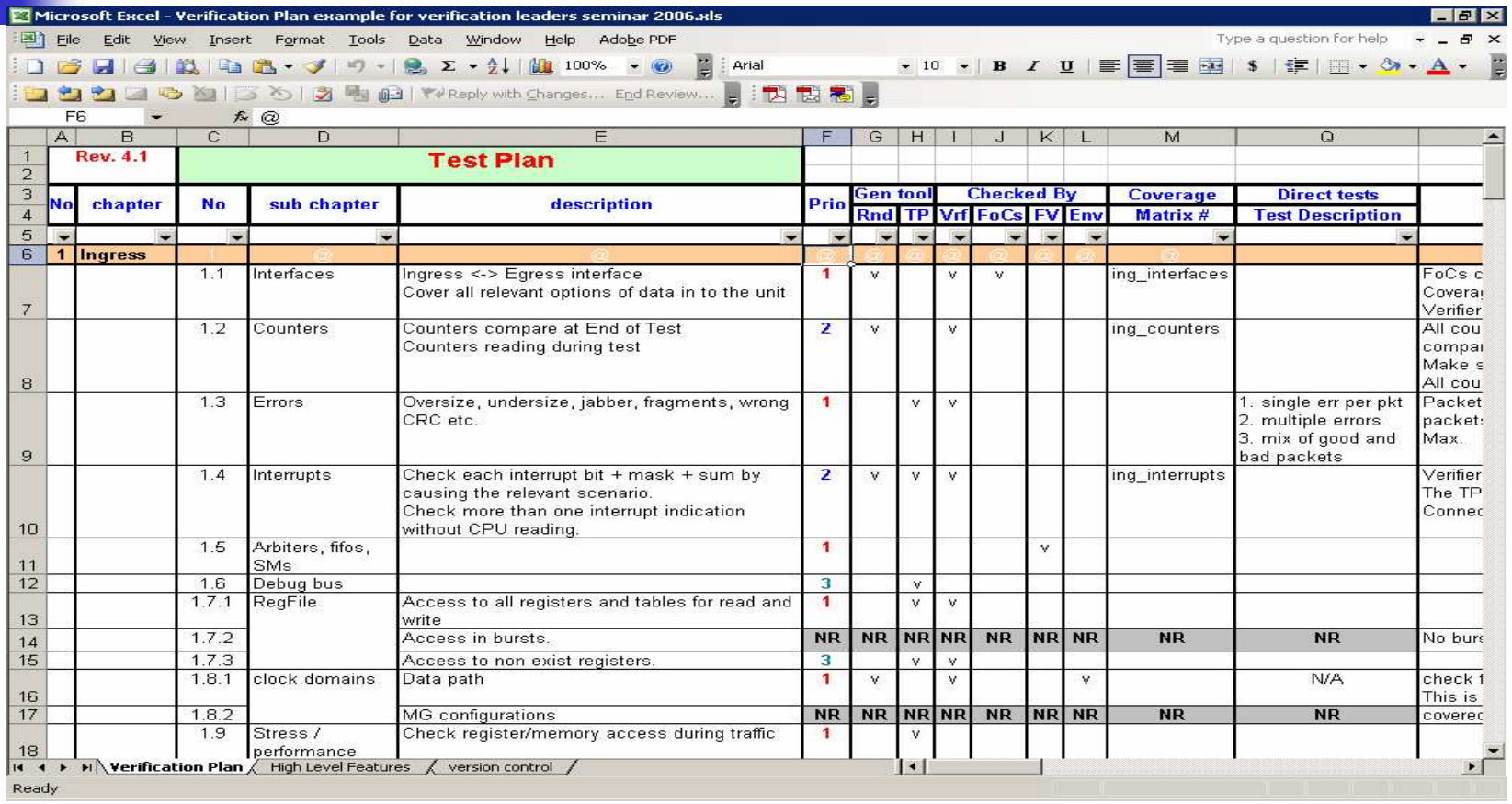
Format

- Different generators / checkers
- High Level / Low level
- Priorities
- Monitoring capabilities

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®

Format Example



Microsoft Excel - Verification Plan example for verification leaders seminar 2006.xls

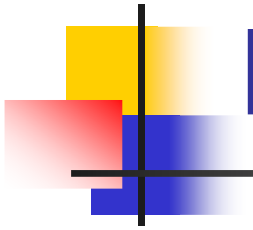
	A	B	C	D	E	F	G	H	I	J	K	L	M	Q	
1		Rev. 4.1			Test Plan										
2															
3	No	chapter	No	sub chapter	description	Prio	Gen tool		Checked By				Coverage	Direct tests	
4							Rnd	TP	Vrf	FoCs	FV	Env	Matrix #	Test Description	
5															
6	1	Ingress													
7			1.1	Interfaces	Ingress <-> Egress interface Cover all relevant options of data in to the unit	1	v		v	v			ing_interfaces		FoCs c Covera Verifier
8			1.2	Counters	Counters compare at End of Test Counters reading during test	2	v		v				ing_counters		All cou compa Make s All cou
9			1.3	Errors	Oversize, undersize, jabber, fragments, wrong CRC etc.	1		v	v					1. single err per pkt 2. multiple errors 3. mix of good and bad packets	Packet packet Max.
10			1.4	Interrupts	Check each interrupt bit + mask + sum by causing the relevant scenario. Check more than one interrupt indication without CPU reading.	2	v	v	v				ing_interrupts		Verifier The TP Connec
11			1.5	Arbiters, fifos, SMs		1					v				
12			1.6	Debug bus		3		v							
13			1.7.1	RegFile	Access to all registers and tables for read and write	1		v	v						
14			1.7.2		Access in bursts.	NR	NR	NR	NR	NR	NR	NR	NR	NR	No burs
15			1.7.3		Access to non exist registers.	3		v	v						
16			1.8.1	clock domains	Data path	1	v		v			v		N/A	check t This is
17			1.8.2		MG configurations	NR	NR	NR	NR	NR	NR	NR	NR	NR	coverec
18			1.9	Stress / performance	Check register/memory access during traffic	1		v							

Verification Plan / High Level Features / version control

Ready

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®



Definition

- When:
  - Must we wait until Arch is closed? (features)
  - Design status - Started? Finished? Other? (components – fifos, arbiters, SMs etc.)
- Who:
  - Arch, µArch, Design, Verification, Other?
  - Split between several engineers

Discretix
embedding security solutions

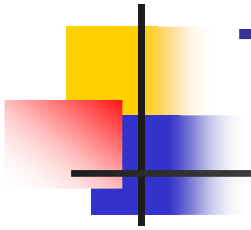
MARVELL®
MOVING FORWARD FASTER®

Review

- Reviewers:
  - Arch, µArch, PL, Design, Verification (All?), Others?
- Risk evaluation
  - Priority = risk x feature necessity
- Depending on the status of:
  - Architecture, Design, Verification
- Re-review
  - Code completion? RTL Freeze? Tape out?

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®



The End

- Thank You!

Discretix
embedding security solutions

MARVELL®
MOVING FORWARD FASTER®