EasyChair Preprint

№ 3572

Switch-Mode based Interposer developed to self-test an MCM without Known Good Dice

Peter Wang

June 7, 2020

SWITCH-MODE BASED INTERPOSER DEVELOPED TO SELF-TEST AN MCM WITHOUT KNOWN GOOD DICE

WANG Shun Shen Peter

JTAG Technologies B.V. Eindhoven, The Netherlands

TamKang University, Taipei, Taiwan

IEEE1149@JTAG.com.sg

Abstract—Improved iJTAG Technology for testing an unknown die at its logic address in lieu of BSDL specified ID code, using a switch-mode based interposer to self-test an MCM without Known-Good-Dice.

Keywords—Interposer, Known-Good-Die, Chip-let, DDR, MCM, self-test, Boundary Scan, substrate, BEOL, JTAG, Bumping, Flip-chip, CPU

I. INTRODUCTION

Unlike the traditional Joint Test Access Group (JTAG) that allows one daisy-chain passes through a TAP port, one at a time, our Switch-Mode Interposer (SMI) self-test system and method permit multi-drop or cascaded ports to branch out a daisy chain into different channels. This eliminates the need for a dedicated ID code for an IC under test, which is required to be present during an Infrastructure test by the JTAG standard.

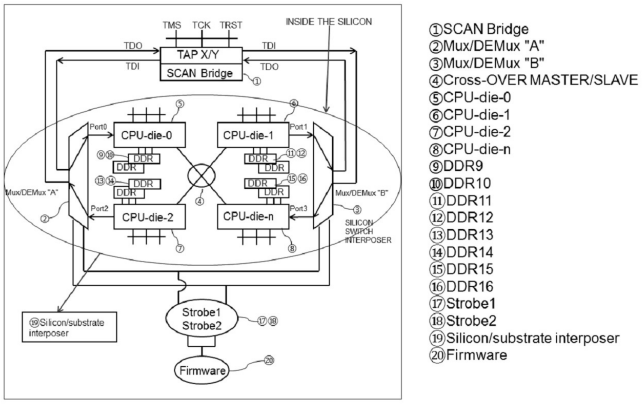
This advanced method will save an IC since it will be tested independently in the absence of a daisy-chain, and enable testing of a faulty device by JTAG's Infrastructure Test that extends to non-boundary scan devices via the "interconnection Test" (i.e. JTAG Interconnection Test).

This system and method are aided by an advanced "architecture", which orders the pairing and looping of TDI/TDO, starting from a "Scan Bridge" based Test Head as the master under control by the JTAG IEEE1149 Controller, passing through multiplexed ports and switched at different directions, clockwise or counter-clockwise, through a mapped address table which put each die under testing at its logic location, automatically.

This innovative method applies a mapped register table in a matrix, given "coordinates" of the dice, which are enabled to provide the pre-defined logic location of the die under testing in the following methods:

1. The Master's Serial Vector Format (SVF);
2. The Slaves' I2C bus, e.g. a CPU;
3. The strobes of "select" and "de-select" at the addressable Latches and MUX/DeMUX;
4. Addressed bus related to Dynamic Random Access Memories such as DDR4.

The following drawing shows the architecture of said method in detail and describes the 20 elements used to build an exemplary system, block by block and node by node.



A. *The Theory of an embedded Tester*

The SMI enables an MCM to self-test itself via a switch-mode based interposer, combined with logic control circuits to drive and sense boundary scan cells at their logic location of their residence dice, and the SMI is configured as a tester to test devices disposed on it, representing a plurality of chiplets and their associated memory chips to be finally integrated as a system in a module.

An important feature of this novel approach is the ability to design a logic map comprised of registers that addresses each logic location, thereby allowing the die to be tested blindly without knowledge of an ID code as required by present testing technologies.
By the same token, for non-boundary devices like the DDRs, which have no ID codes any way, they can be tested functionally in a similar manner. Depending on their addressable buss, once their related CPUs are called upon, the DDRs are identified as extensions through Data, Address and Control buss, thereby allowing the bus signals to be tested functionally.

In a complex silicon interposer based MCM, a faulty die is often induced by thermal stress and related environmental issues such as, for example, flip chip based bumping process like C4, etc.; cold or hot soldering generating open or shorts on stack-up dice; and discontinuity and/or cross-talks during high speed transmissions.

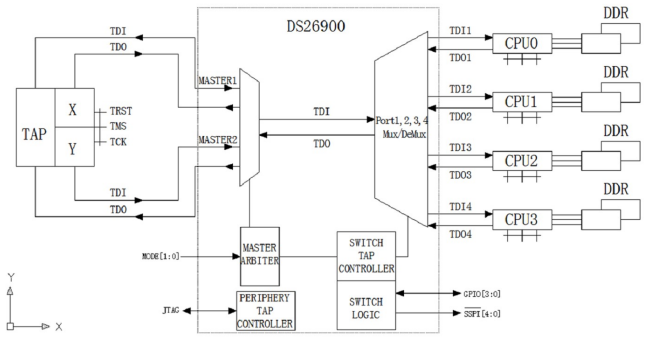
We have solved these problems on a silicon interposer level, and could improve yield of MCMs. There are three critical elements used for building this advanced SMI system: 1) a transparent scan bridge to instruct TAP X/Y of test heads, 2) Mux/DeMux to selectively steer these test heads into various quadrant ports, and 3) through a cross-over bus mastered by I2C protocol to switch new path towards each slave, i.e., devices under test.

The MCM are notoriously difficult to debug if an unknown die is present because (1) there is no label on each chip, (2) bare dice are made to flip-chip with hidden bumping, and (3) they are blindly assembled together.

This new and innovative test method has developed a logic system for dice to address their own locations in an MCM through an embedded registers system to guide the TDI/TDO test signals to behave like a pair of index fingers pointing to the die under test at specific port and address. This built-in steering system embedded in a passive or active interposer provides a novel approach for the Design For Testing (DFT) at die level and which uses module based schematics of which required netlist to interconnect all pins and micro bumps of dice in an MCM.

B. The SMI Test Engine

This new method is achieved by selectively steering signals through a pair of TDI/TDO instead of all five TAP signals (TDI, TDO, TCK, TMS and TRST) to a die under test at its corresponding local locations according to a look-up table through multiplexing and de-multiplexing plural TAP ports. These ports are switched at a cross-over point, either in a forward or reverse direction in two-way traffic, as explained in the following diagram:



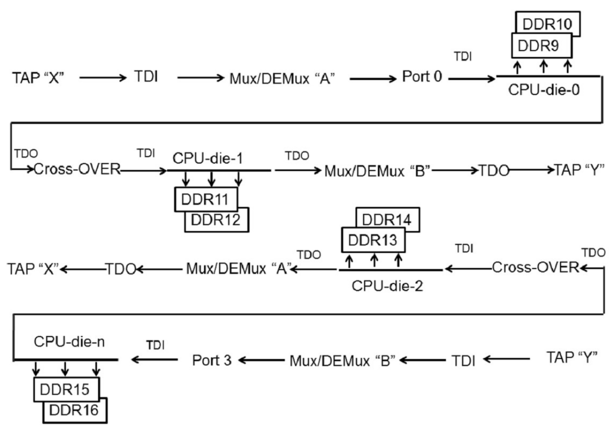
We use a commercially available DS26900 chip to mux the master channels exchanging from 4:2 to 2:8 slaves channels, controlled by GPIO and SSPI terminals as a self-guide system which is switched by I2C protocol through Serial Data Line (SDA) and Serial Clock Line (SCL) bus lines.

C. Flow-chart of Test Progress

The First Step is to research an advanced architecture in lieu of the traditional approach that requires a Boundary Scan ID code to indicate which chip is under test. An MCM only carries bare dice without noticeable die markers, thereby making test paths dependent Daisy Chains invaluable during trouble-shooting and diagnostics. This is made worse when a boundary scan required "Interconnections Test" is only partially performed by the "Scan-out Test", rendering the bad die indeterminate as to its causes of failures, and in the case of bumped dice the failures become undetectable in short of "scan-in test". This occurs because a silicon interposer has no daisy chain to support complicated "Interconnection Test" through scan in and out of boundary cells.

Unless a new architecture is developed by using JTAG based Netlist (so called JTN), this method allows one to test each die individually through its I/O pins associated

boundary scan cells, to thereby drive and sense each die by Test Input and Output Data (TDI/TDO) only and not through its control signals (TMS, TCK and TRST) in parallel bus. This method reduces routines greatly to relax the limitations on silicon layouts, floor planning as well as logic gating if sense in and drive out all five TAP signals to each die as used to be, in the case of a multiprocessor integrated with plural chiplets and DDRs shown in the chart below:



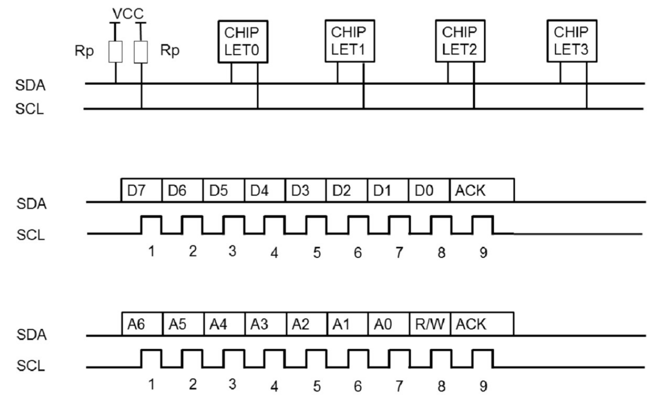
D. Registers based Map to locate each die assigned Logic Address per I2C Bus

The chart below illustrates a double-looping indexed test head "TDI/TDO" running in two directions, either clockwise or counterclockwise starting at the "Scan Bridge" and are channelized by Tap Access Port (TAP) in a transparent X/Y path, branching out two separate ports, and switched by a pair of "Mux/DeMux" to locate a chosen port. Once the test signal enters the networked chiplets, it is switched to an individual die and then guided through a cluster of dice, to avoid a bad one, and thus selectively to reach a second chiplet in accordance to their register based control logic as such:

DS26900 X							DS26900 Y						
EREQ	TMREQ1	TMREQ2	GPIO4	GPIO3	GPIO2	GPIO1	EREQ	TMREQ1	TMREQ2	GPIO4	GPIO3	GPIO2	GPIO1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1

PCA9500								SN74LVC2T45								TAP X/Y	
IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7	1	2	3	4	5	6	7	8		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	CPU0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	CPU2	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		CPU1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		CPU3
0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	CPU0-CPU1	CPU3-CPU2
1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	CPU0-CPU3	CPU1-CPU2
1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	CPU1-CPU0	CPU2-CPU3
1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	CPU3-CPU0	CPU2-CPU1

The chart below further presents how a standard I2C bus driven by "Master and Slaves" Protocol in search of a selected die, and the I2C bus uses only two bidirectional open collector or open drain lines, so called Serial Data Line (SDA) and Serial Clock Line (SCL), and pulled up with resistors. The bus has two roles for the nodes, specially Master and Slave, herewith the chiplets 1-4 on the top row:



E. Scan Bridge Evolved "Transparent Test Head" Technology

Many modern communication and networking systems incorporate a system-wide IEEE 1149.1 (JTAG) test bus. The test bus not only enables a comprehensive, life-cycle approach to system test, but it also offers a number of additional benefits to the system designer.

The utility of the JTAG bus continues to expand beyond testing. JTAG is now used for emulation, memory programming, and configuration of CPLDs or FPGAs. These applications of the JTAG bus are well supported by the industry, including Design for Testing (DFT) an MCM.

JTAG bus infrastructures continue to become more extensive, expanding to many chip-lets on a silicon based interposer and to multiple DDRs within a multiple processors based system. The need to manage the JTAG bus becomes apparent as the bus infrastructure expands. grouping devices with similar technologies into smaller dice like chiplets, local scan chains can reduce complexity and improve debug and fault isolation. Partitioning sets of components into local scan chains maximizes access for speed-critical applications, such as high speed CPU and DDR as a "chip-set".

Computing Systems with multiple chip-lets and DDRs populated on an active interposer are designed to extend the JTAG infrastructure on each chip-let by extending the JTAG test bus throughout the interposer. Multiple dice can share the test bus when each chip-set (chip-let

and DDRs) utilizes a JTAG interface device that enables multi-drop addressing using the controlled Test Accessible Ports.

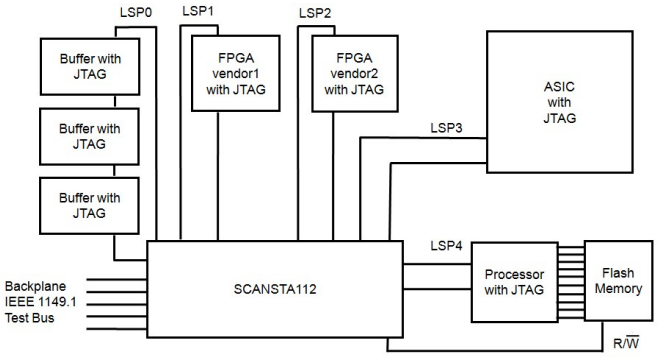
Texas Instruments SCANSTA111 and SCANSTA112 (STA11x) devices enable partitioning of complex scan chains. The SCANSTA111 and SCANSTA112 support multidrop addressing, and manage up to three or seven local scan chains, respectively. The STA11x devices are commonly used for isolating components on a circuit board. This is particularly useful for configuring Chip-lets and DDRs based "chip-sets".

There are three basic ways to get to the STA11x devices: Transparent Stitcher Mode, Transparent Scan Bridge Mode, and Normal Scan Bridge Mode.

Transparent stitcher mode (STA112 feature applied only) is activated by way of external pins, and does not require changes to the SVF. This mode does require external hardware control. Setting the SB/S pin to 0 selects the stitcher mode; setting the TRANS pin to 1 puts the device into transparent mode, and the LSP select pins to the appropriate Master abd Slaves channels:

- Master Mode-node that generates the clock and initiates communication with slaves,
- Slave Mode-node that receives the clock and responds when addressed by the master,
- Only the SDA line changes direction during acknowledge bits; the SCL is always controlled by The master

In the case of SMI, the test head run by JTAG controller to send TDI and TDO signals in transparent mode through TAP Ports acting as the master, typical controller hardware organized as follows:



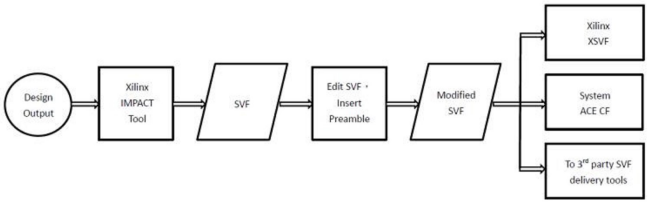
This IEEE1149 based JTAG controller gives control and command, verified by Instruction Register (IR) and Data Registers (DR), to set TDI and TDO "transparently" as X and Y, keyed by "1sp0" and "1sp1" respectively, executed, showing the following examples:

TRST ON;
TRST OFF;
S I R 8 TDI (11) ; ! Address Scan-Bridge
S I R 8 TDI (A0) ; ! Load instruction to enable transparent mode for LSP0
S I R 8 TDI (a5) ; ! Verify SIR
S D R 8 TDI (5a) ; ! Verify SDR
S I R 8 TDI (C3) ; ! Try to load GOTOWAIT in Scan-Bridge
D S R & TDI (5a) ; ! Verify that Scan-Bridge did not recognize GOTOWAIT
! Now TDIB→ "1sp0"→TDO-B
TRST ON;
TRST OFF;
S I R 8 TDI (11) ; ! Address Scan-Bridge
S I R 8 TDI (A1) ; ! Load instruction to enable transparent mode for LSP1
S I R 8 TDI (a5) ; ! Verify SIR
S D R 8 TDI (5a) ; ! Verify SDR
S I R 8 TDI (C3) ; ! Try to load GOTOWAIT in Scan-Bridge
S D R 8 TDI (5a) ; ! Verify that Scan-Bridge did not recognize GOTOWAIT
! Now TDIB→ "1sp1"→TDO-B

Test Head used by SMI, written in SVF file informs the JTAG Controller to run the Master and Slave on a I2C bus towards each specific ports, and the SVF file specify paired TDI/TDO as SPSR in firmware code as shown in the following Registers:

```
! Register Name: SPSR
! Register Description:  5-Bit Secondary Port Selection Register

! Bit#  | 7 6 5 4    3     2     1     0     |
! Name  | — — —  SSP[4]  SSP[3]  SSP[2]  SSP[1]  SSP[0]  |
! Reset | — — — 0     0     0     0     0     |
! Bits 4 to 0: Secondary Port Selection (SSP[4:0])

! Selection of the Secondary Port
! Selection of the secondary port ("slave") is accomplished by writing a 5-bit address into the
Secondary Port
! Selection Register (SPSR). Due to the star configuration, only one port can be selected at a time.
Ports that are not
! detected as being present by sensing the pullup on the secondary port's TMS pin can still be
selected, and the
! signals will be sent to that port.
! This 5-bit secondary port selection address is complemented and used to generate the selected
slave port indicator
! bits (SSPI[4:0]). These bits can be used as a visual indicator as to which slave port has been
selected.
! Once communications with a secondary port has been completed, the Secondary Port Selection
Register (SPSR)
! should be set to all zeros. If not, the selected port address will not respond to the DPDV bit of the
Device
! Configuration Register (DCR). This is true if an active master is present or not.
!
FREQUENCY 1.0E+06 HZ;
STATE IDLE;
RUNTEST IDLE 1000 TCK ENDSTATE IDLE;

SIR 5 TDI (02)       ! spsr 00010
     TDO (02)
                        MASK (1F);
SDR 8 TDI (01)       ! XXX  00001- port1 00010- port2 00011- port3 00100- port4
     TDO   (00)
                        MASK(FF);

RUNTEST 1000 TCK;
```

and to execute the steps of the following diagram progressively by a modified Serial Vector Form (SVF):



More specifically, four chip-let based CPU in U1, U2, U3 and U4 selected by 8-bit data requests lines in 32 addressable locations, are shown below:

CTRL BUS	DQM0	DQM1	DQS0	DQS1	DQM2	DQM3	DQS2	DQS3	DQM4	DQM5	DQS4	DQS5	DQM6	DQM7	DQS6	DQS7
U1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
U2	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
U3	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	0
U4	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1

F. Extension of Dynamic Memories (DDR) Under Testing

In this application, we have emulated 8 pairs of DDRs, in extension to 4x CPU interconnected via Data, Address and Control buss, bench-marked as follows:

A) Master Controller handles the DDR associated bus lines in extension to CPU related Interconnections, specially assigned to their address control bits as following table defined bus lines:

- 16-bits address buss
- 64-bits data buss
- 6-bits selection bus
- 6-bits common buss

U1

ADDRESS BUS	DATA BUS	CTRL BUS
MA0	DATA0	DQS0
MA1	DATA1	DQS0_N
MA2	DATA2	DQS1
MA3	DATA3	DQS1_N
MA4	DATA4	DQM0
MA5	DATA5	DQM1
MA6	DATA6	
MA7	DATA7	RAS
MA8	DATA8	CAS
MA9	DATA9	WE
MA10	DATA10	CS
MA11	DATA11	CKE
MA12	DATA12	RESET
MA13	DATA13	
BA0	DATA14	
BA1	DATA15	
BA2		

U3

ADDRESS BUS	DATA BUS	CTRL BUS
MA0	DATA32	DQS4
MA1	DATA33	DQS4_N
MA2	DATA34	DQS5
MA3	DATA35	DQS5_N
MA4	DATA36	DQM4
MA5	DATA37	DQM5
MA6	DATA38	
MA7	DATA39	RAS
MA8	DATA40	CAS
MA9	DATA41	WE
MA10	DATA42	CS
MA11	DATA43	CKE
MA12	DATA44	RESET
MA13	DATA45	
BA0	DATA46	
BA1	DATA47	
BA2		

U2

ADDRESS BUS	DATA BUS	CTRL BUS
MA0	DATA16	DQS2
MA1	DATA17	DQS2_N
MA2	DATA18	DQS3
MA3	DATA19	DQS3_N
MA4	DATA20	DQM2
MA5	DATA21	DQM3
MA6	DATA22	
MA7	DATA23	RAS
MA8	DATA24	CAS
MA9	DATA25	WE
MA10	DATA26	CS
MA11	DATA27	CKE
MA12	DATA28	RESET
MA13	DATA29	
BA0	DATA30	
BA1	DATA31	
BA2		

U4

ADDRESS BUS	DATA BUS	CTRL BUS
MA0	DATA48	DQS6
MA1	DATA49	DQS6_N
MA2	DATA50	DQS7
MA3	DATA51	DQS7_N
MA4	DATA52	DQM6
MA5	DATA53	DQM7
MA6	DATA54	
MA7	DATA55	RAS
MA8	DATA56	CAS
MA9	DATA57	WE
MA10	DATA58	CS
MA11	DATA59	CKE
MA12	DATA60	RESET
MA13	DATA61	
BA0	DATA62	
BA1	DATA63	
BA2		

Through U1 to U4, these DDRs with respect to their associated CPU, forming a group of "chip-sets", there are four groups of such chip-sets to be bench-marketed by Boundary Scan Test (BST).

G. JTAG Technologies' system test software applied to SMI

The following test result demonstrates the effectiveness of this bench-marked SMI system; there are 4 CPUs in connection with 8 DDRs in this case, and they are executed in sequence per die base throughout the MCM, showing no error as a result of all dice are good even though they are represented anonymously from 0 to 3 for both chip-lets and DDR4 serially.

Nr.	Type	Oper	Design	Item	Show	Status
1	Pld	Dnld	00_Chiplet0_select	00_Chiplet0_select	Stat	Passed
2	Test	Exec	01_Infra_Chiplet0	infra	Stat	Passed
3	Pld	Dnld	00_Chiplet1_select	00_Chiplet1_select	Stat	Passed
4	Test	Exec	01_Infra_Chiplet1	infra	Stat	Passed
5	Pld	Dnld	00_Chiplet2_select	00_Chiplet2_select	Stat	Passed
6	Test	Exec	01_Infra_Chiplet2	infra	Stat	Passed
7	Pld	Dnld	00_Chiplet3_select	00_Chiplet3_select	Stat	Passed
8	Test	Exec	01_Infra_Chiplet3	infra	Stat	Passed
9	Pld	Dnld	00_Chiplet0_select	00_Chiplet0_select	Stat	Passed
10	Test	Exec	01_Infra_Chiplet0_Chiplet1	infra	Stat	Passed
11	Test	Exec	01_Infra_Chiplet0_Chiplet2	infra	Stat	Passed
12	Test	Exec	01_Infra_Chiplet0_Chiplet3	infra	Stat	Passed
13	Pld	Dnld	00_Chiplet1_select	00_Chiplet1_select	Stat	Passed
14	Test	Exec	01_Infra_Chiplet1_Chiplet0	infra	Stat	Passed
15	Test	Exec	01_Infra_Chiplet1_Chiplet2	infra	Stat	Passed
16	Test	Exec	01_Infra_Chiplet1_Chiplet3	infra	Stat	Passed
17	Pld	Dnld	00_Chiplet2_select	00_Chiplet2_select	Stat	Passed
18	Test	Exec	01_Infra_Chiplet2_Chiplet0	infra	Stat	Passed
19	Test	Exec	01_Infra_Chiplet2_Chiplet1	infra	Stat	Passed
20	Test	Exec	01_Infra_Chiplet2_Chiplet3	infra	Stat	Passed
21	Pld	Dnld	00_Chiplet3_select	00_Chiplet3.Select	Stat	Passed
22	Test	Exec	01_Infra_Chiplet3_Chiplet0	infra	Stat	Passed
23	Test	Exec	01_Infra_Chiplet3_Chiplet1	infra	Stat	Passed
24	Test	Exec	01_Infra_Chiplet3_Chiplet2	infra	Stat	Passed
25	Pld	Dnld	00_Chiplet0_select	00_ Chiplet0_ _select	Stat	Passed
26	Test	Exec	02_Interconnect_Chiplet0_Chiplet1	02_Interconnect_Chiplet0_Chiplet1	Stat	Passed
27	Test	Exec	02_Interconnect_Chiplet0_Chiplet2	02_Interconnect_Chiplet0_Chiplet2	Stat	Passed
28	Test	Exec	02_Interconnect_Chiplet0_Chiplet3	02_Interconnect_Chiplet0_Chiplet3	Stat	Passed
29	Pld	Dnld	00_Chiplet1_select	00_Chiplet1_select	Stat	Passed
30	Test	Exec	02_Interconnect_Chiplet1_Chiplet0	02_Interconnect_Chiplet1_Chiplet0	Stat	Passed
31	Test	Exec	02_Interconnect_Chiplet1_Chiplet2	02_Interconnect_Chiplet1_Chiplet2	Stat	Passed
32	Test	Exec	02_Interconnect_Chiplet1_Chiplet3	02_Interconnect_Chiplet1_Chiplet3	Stat	Passed
33	Pld	Dnld	00_Chiplet2_select	00_Chiplet2_select	Stat	Passed
34	Test	Exec	02_Interconnect_Chiplet2_Chiplet0	02_Interconnect_Chiplet2_Chiplet0	Stat	Passed
35	Test	Exec	02_Interconnect_Chiplet2_Chiplet1	02_Interconnect_Chiplet2_Chiplet1	Stat	Passed
36	Test	Exec	02_Interconnect_Chiplet2_Chiplet3	02_Interconnect_Chiplet2_Chiplet3	Stat	Passed
37	Pld	Dnld	00_Chiplet3_ select	00_Chiplet3_select	Stat	Passed
38	Test	Exec	02_Interconnect_Chiplet3_Chiplet0	02_Interconnect_Chiplet3_Chiplet0	Stat	Passed
39	Test	Exec	02_Interconnect_Chiplet3_Chiplet1	02_Interconnect_Chiplet3_Chiplet1	Stat	Passed
40	Test	Exec	02_Interconnect_Chiplet3_Chiplet2	02_Interconnect_Chiplet3_Chiplet2	Stat	Passed
41	Pld	Dnld	00_Chiplet0_select	00_Chiplet0_select	Stat	Passed
42	Test	Exec	03_DDR4_SDRAM_U2_0_Chiplet0	03_DDR4_SDRAM_U2_0_Chiplet0	Stat	Passed
43	Test	Exec	03_DDR4_SDRAM_U2_1_Chiplet0	03_DDR4_SDRAM_U2_1_Chiplet0	Stat	Passed
44	Pld	Dnld	00_Chiplet1_select	00_Chiplet1_select	Stat	Passed
45	Test	Exec	03_DDR4_SDRAM_U2_2_Chiplet1	03_DDR4_SDRAM_U2_2_Chiplet1	Stat	Passed
46	Test	Exec	03_DDR4_SDRAM_U2_3_Chiplet1	03_DDR4_SDRAM_U2_3_Chiplet1	Stat	Passed
47	Pld	Dnld	00_Chiplet2_select	00_Chiplet2_select	Stat	Passed
48	Test	Exec	03_DDR4_SDRAM_U3_0_Chiplet2	03_DDR4_SDRAM_U3_0_Chiplet2	Stat	Passed
49	Test	Exec	03_DDR4_SDRAM_U3_1_Chiplet2	03_DDR4_SDRAM_U3_1_Chiplet2	Stat	Passed
50	Pld	Dnld	00_Chiplet3_select	00_Chiplet3_select	Stat	Passed
51	Test	Exec	03_DDR4_SDRAM_U3_2_Chiplet3	03_DDR4_SDRAM_U3_2_Chiplet3	Stat	Passed
52	Test	Exec	03 DDR4_SDRAM_U3_3_Chiplet3	03 DDR4_SDRAM_U3_3_Chiplet3	Stat	Passed

II. Summary Of SMI Required Logic Deigns And Fabrication

A. Interposer related Logic Designs

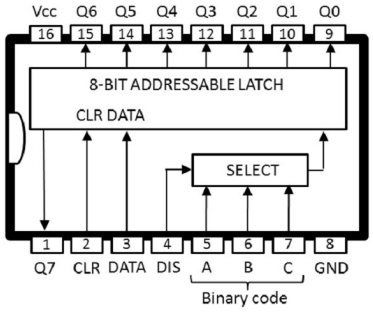
SMI required logic designs will ask for gates-level Mux Design in terms of how many transistors needed, to reduce complexity involved in making active interposer, in the case of a simple non-restoring Mux, it may be limited to two transmission gates as few as 4 transistors; for examples:

- choose one of four inputs using “select” of two level of 2:1 MUX or four transistors;
- Of 4:1 MUX choosing one of 4 inputs by using two “selects” at two levels of 2:1 Muxes or eight transistors of back to back connected latches in four “ selects”;
- When CLK = 1, the latch is transparent, data flows to output data “Q” like a buffer, when CLK = 0, the latch is opaque and output Q held as its old value and independent to data “D”, i.e, the MUX chooses D or old Q;
- Master and Slave operating when CLK rises, D is copied to Q, and all other times, Q holds its value, i.e., a positive edge-triggered Flip-Flop (FF), to build the Master and Slave flip-flop;
- CMOS SR Latch for NOR Gate version, the NOR-based SR Latch contains the basic memory cell (back-back inverters into two NOR gates to allow setting the state of Latch;
- CMOS SR Latch for NAND version, a CMOS S-R latch built with two 2-input NAND gates, and the circuit responds to active low S and R inputs;
- Clocked CMOS J-K Latch of NAND version, by using feed-back from output to input to eliminate indeterminate state when both S and R are high;
- When a D Latch implemented at the gate level, by simply utilizing a NOR-based S-R Latch to connect D to input S as well as D to input R with an inverter, and thus the D Latch is normally implemented with “Transmission Gate (TG) Switch” in this case.

B. Interposer-related Function Designs

1) CMOS Addressable Latches

CMOS 4099B as an 8-bit addressable latch device, it is made of functional diagram, address chart, and function table shown in figures below:



BINARY CODE			LATCH ‘Q’
C	B	A	SELECTED
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Address Table

DIS	CLR	Addressed Latch	Unaddressed Latch	Mode
0	0	Follows Date	Holds Previous Data	Addressable Latch
1	0	Holds Previous Data	Holds Previous Data	Memory
0	1	Follows Data	Reset to ‘0’	Demultiplexer
1	1	Reset to ‘0’	Reset to ‘0’	Clear

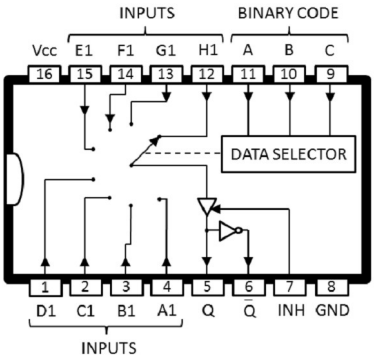
Function Table

This particular IC presents a logic building block to serve SMI through firmware selection of different modes like a steering wheel, to selectively hold the previous data, or to follow the data, and reset the date to “0”. Conveniently, the DIS pin (select disable) acts like a terminal of clock signal, and must be pulled “low” to

load the binary selected code into latch, otherwise it will act like an addressable latch when CLR is low, or as a demultiplexer when CLR is high.

2) CMOS based Multiplexer

Another building block related to SMI has been the application of CMOS 74HC152, an 8-input data selector and/or multiplex, shown in the figure below:



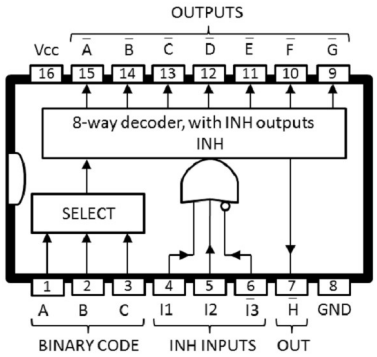
BINARY CODE			INPUTS
C	B	A	SELECTED
0	0	0	A1
0	0	1	B1
0	1	0	C1
0	1	1	D1
1	0	0	E1
1	0	1	F1
1	1	0	G1
1	1	1	H1

DATA SELECTION TABLE

This device is capable of handling a single "8-way switch" and that can be disabled by biasing the INH terminal high; the switch positions can be selected by applying a 3-bit binary codes to connect the device as an 8-way input selector, which outputs whatever switch is selected via the selection code on the right side table.

3) CMOS based DeMultiplexer

CMOS device 74HC138, 8-way decoder/demultiplexer device. Its functionalities are shown in the figure below; this device can be used as a demultiplexer of SMI by applying one of the active-low INH terminals as the data input, with the other two INH inputs disabled, SMI functional building block shown in figure below in reference to selection code on the right side table;



BINARY CODE			INPUTS
C	B	A	SELECTED
0	0	0	Ā
0	0	1	B̄
0	1	0	C̄
0	1	1	D̄
1	0	0	Ē
1	0	1	F̄
1	1	0	Ḡ
1	1	1	H

DECODING TABLE

III. Investigation Of SMI Required Wafer Back-End Fab

A. BEOL Wiring Process for Logic Switch and Clock Circuits associated Fabrications

The terms FEOL (Font End Of the Line) and BEOL (Back End Of the Line) originated at SEMATECH several years ago as an attempt to describe the growing difference between processes for logic circuits and those for DRAM circuits, SEMATECH's SRAM process are very similar to logic processes; however in this paper, we are limited to logic switch and clock processed by BEFL without considering a driver and buffer circuits that generate or regenerate the output signals for microprocessor and dynamic random access memory (DDR) which are normally fabricated by the FEOL, and there are two most updated BEFL process currently available to fabricate SMI. Some are listed as follows:

- INTACT: "Active Interposer Technology for chiplet-based advanced 3D system architecture" in reference to published 2019 IEEE 69th Electronic Components and Technology Conference (ECTC). [6]
- TSMC's Homogeneous CoWoS (Chip-on-Wafer-on-Substrate) by S.Y. Hou and NTU's Chip on Wafer (COW) by M. H. Liao

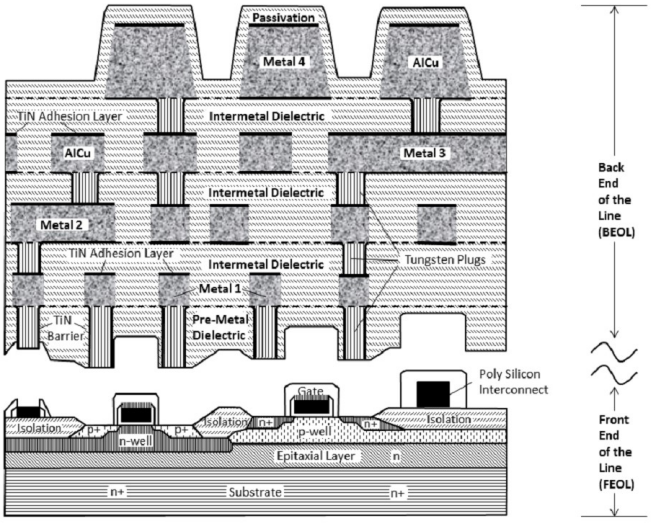
B. The Weight Between FEOL and BEOL

The following drawing presents the insignificance of front-end-of-line (FEOL) and emphasizes back-end-of-line (BEOL) intensive circuits to justify switch-mode required logic and gates in semiconductor related transistor designs. These seldom contain passives like resistors, capacitors or locally interconnected signals, but only transistors which are BEOL intensive circuits to switch and multiplex boundary scan TAP signals; wherein referred to VLSI Research's BEOL Wiring Process for CMOS Logic.[3]

C. Built-in Active Interposer with Switch and MUX Circuits

To further explore the possibility and to build common logic of switch and MUX, we are looking into available integration platforms such as noted INTACT system which may be catered to insignificant FEOL, but put emphases on CMOS born BEOL, prototyping flow as shown in following cross-section of a typical active interposer required wafer fab:

1) CMOS 65nm front end of line (FEOL)
2) through silicon vias (TSV) in the middle
3) CMOS BEOL with 7 metal layers ad AL pads
4) Cu pillar above AL pads
5) Temporary wafer bonding and silicon thinning
6) Cu nail reveal the insulation and opening
7) Redistribution layer (RDL) and passivation
8) Solder bumps



IV. CONCLUSIONS

This paper has described our novel SMI self-testing of an MCM. It innovatively applies a look-up table to address a registered die under testing boundary scan devices in extension to non-boundary scan ICs such as the DDRs, and still in compliance with the Joint Test Access Group (JTAG) standard as per IEEE1149 required protocols.

To achieve the set objectives of SMI, we have developed a brand new architecture that is indifferent to the traditional daisy-chain based infrastructure that links up ID codes in order to recognize each device and which is serially connected through Test Data Input (TDI) and Test Data Output (TDO) along with Test Mode Select (TMS), Test Clock (TCK) and Test Reset (TRST) bused in an arrangement of parallel. Instead, we only use TDI and TDO as a pair of index fingers guided through a steering mechanism that crosses a group of control logics including a plurality of multiplexer and de-multiplexer, to differentiate their master and slave at specific ports, using firmware to redirect the path toward a targeted die, which may be a specific chiplet-based CPU in extension to its cache memories, tested by JTAG standard software,

The most troublesome dice have been the flip-chips based DDRs, since they are stacked up in decks by micro-bumps assembly, and it is difficult to dissipate their heat to the adjacent power hustler, i.e., CPU, failures occasionally occur at micro-bumps due to heat transfer induced fatigue and cracks.

In case the CPU has passed the boundary scan test but the DDR does not, this SMI system allows trouble-shooting failure modes per net for the devices, and to debug the failures such as open, short, bridging, etc. related to the pin. On the other hand, if the DDRs are good according to the test, then it is interpreted to mean the CPU is able to emulate data, address and control signals to and from the DDRs with respect to their connected boundary scan cells.

ACKNOWLEDGEMENT

The work was supported by the JTAG Technologies B.V. founded in 1993, and special thank for its founder, Mr. Peter van den Eijnden who has endorsed IEEE1149 ranging from the smallest MCM to largest computing servers as far as the latest autonomous vehicles related ADAS electronics.

The author expresses appreciation to Mr. Wang Yin-Tien, Department Head of Mechanical and Electrical Engineering, and Mr. Yang Wei-Bin, Associate Professor and Chairman, Department of Electrical and Computer Engineering, of Tamkang University, Taipei, Taiwan for their encouragement and knowledgeable inputs in Research and Development.

REFERENCES

The paper numbers citations non-consecutively within brackets [1]. The sentence punctuation follows the bracket [2], referred simply to the reference number, as in [3], etc.

[1] Texas Instruments, “AN-1340 Simplified Programming of Xilinx Devices Using a SCANSTA111/112 JTAG Scan Chain Mux”, Dallas, Texas
[2] Texas Instruments, “ Understanding the I2C Bus,” Application Report, June 2015
[3] VLSI Research Inc. “BEOL Wiring Process for CMOS Logic,” 1754 Technology Drive, Ste 117, San Jose, CA February, 1995
[4] Maxim -IC, “DS26900 JTAG Multiplexer/Switch”, 19-5747, Rev 1

[5] R M. Marston, “Modern CMOS Circuit Manual,” Newnes, An imprint of Butterworth-Heinemann Ltd, Linacre House, Jorden Hill, Oxford, 1996
[6] Perceval Coudrain, etc., “Active interposer technology for chiplet based advanced system architecture”, 2019 69th Electronic Components and Technology Conference (ECTC), pp 569-578