Figure 1 shows a bottom die and the IEEE 1149.1 TAP interface necessary for a 3D die. The interface includes the TCK, TMS, TDI and PTDO on the primary side of the die. On the secondary side, TCK, TMS STDO and STDI are provided to access the next die in the stack. Note that two falling edge latches are used one for PTDO and one for STDO such that the output 1149.1 scan changes state on the falling edge of TCK as per 1149.1. The difference here is two outputs are present. TMS is gated until it is enabled by the UP3D instruction. STDI is registered on the rising edge of TCK from the prior die in order to improve timing. STDI is then followed by the primary falling edge latch.

Figure 1 shows a bottom die of a 3DIC when the EXTEST instruction is loaded. The boundary register includes the SEGSEL S1 and the segment in DOMAIN Vcc_io. The path is highlighted by the heavy weight line. The example illustrates the capability of defining the segment to be powered by a power-pin which brings in the voltage for the I/O. The definition can then be used in diagnostics to identify faults such as when the VCC_IO is open. PCB level interconnect tests which fail a particular group of I/O can identify to the user to check the VCC_IO pin for opens, thereby minimizing false indictments of good components due to a simple manufacturing defect. The isolated power domains shown include circles at the interface which represent the additional power-domain circuitry needed for signals which cross into controllable power-domains. IEEE 1149.1-2013 was designed to work hand-in-hand with IEEE 1801-2013 which defines the power intent of a design.

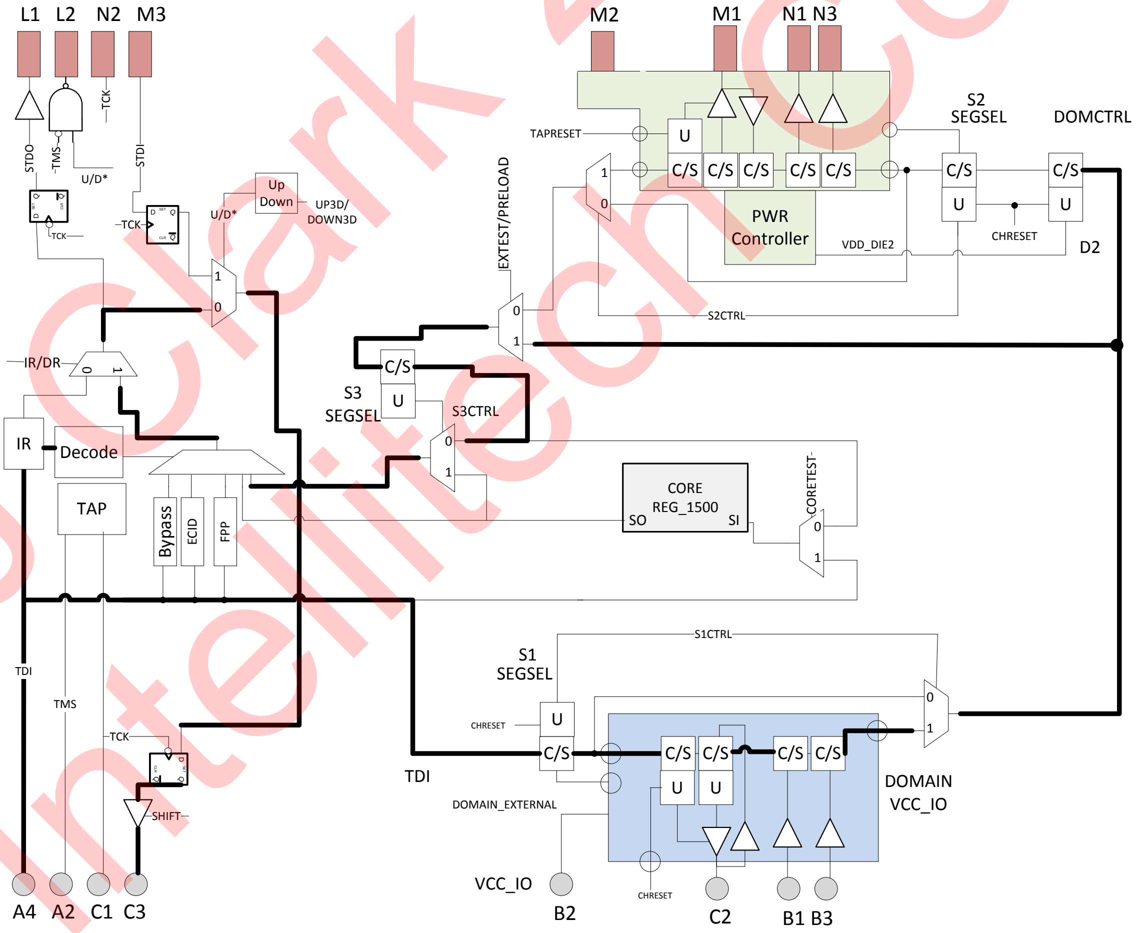


Figure 1. Bottom Die in EXTEST

Not shown in the diagram is the gating of IEEE 1149.1/IEEE 1500 Capture, Update and Shift. The IEEE 1149.1-2013 standard now shows both gated clock and synchronous clock techniques which map to IEEE 1500 style registers. Each instruction (see Figure 2) provides gating of the signals.

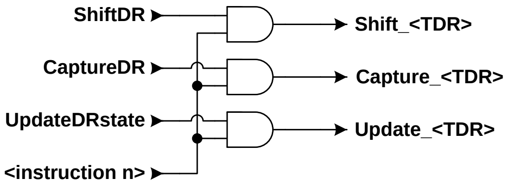


Figure 2. Instruction gating states for TDR

Each SEGSEL or SELECTFIELD control is also responsible for gating the signals locally when the segment is not included in the path (See Figure 3). The SEGSEL (Segment Select Cell) can observe the readiness of an excludable segment such that the segment is never enabled or shifted through when a condition such as the power within the domain the segment resides is turned off. 1149.1-2013 also supports selectable scan segments (*e.g.* segments controlled by an IEEE 1500 WIR).

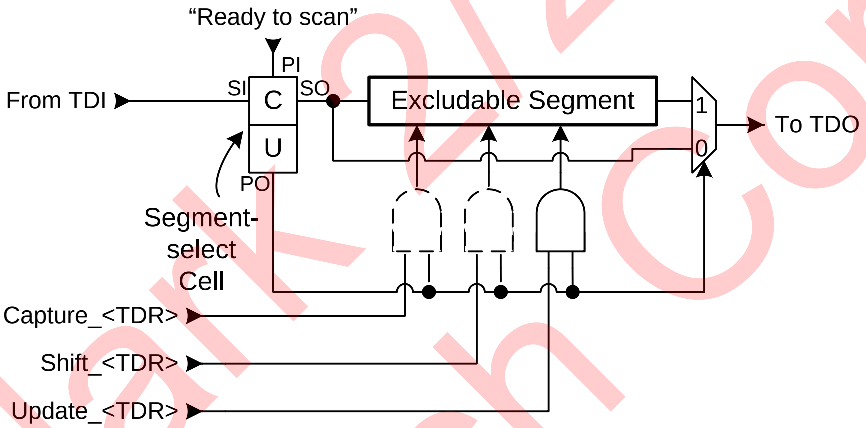


Figure 3. Local gating for Segment.

Figure 1 uses C/S to represent a Capture/Scan flop and U to indicate an Update register is present and required for success. Figure 4 shows a generic representation of the IEEE 1149.1-2013 TDR (Test Data Register) bit using the non-gated clock approach and IEEE 1500 compatible interface. The dashed lines represent options for a TDR bit. Capture input, reset, update flops are all optional and can be described via BSDL. Additional cells are described which augment fault coverage such as self-monitoring cells (MON) and self-resetting cells (PULSE1/PULSE0) which reduces the number of scans needed when transferring data from the TCK domain (scan register) into registers (data registers) in an alternate clock domain such as the mission mode clock. Note also that C/S flops can be shared (using the keyword SHARED) with functional operation rather than dedicated flops. There are many more option TDR register bit types available which aren't fully described in the example BSDLs and figures.

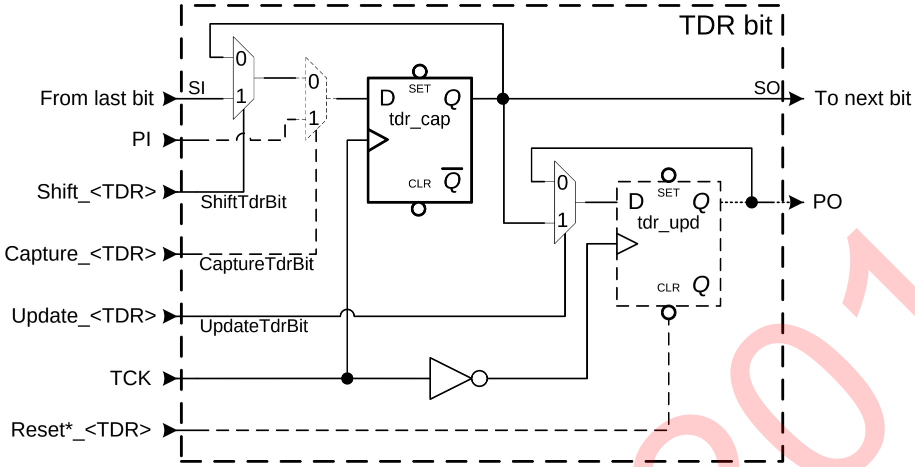


Figure 4. 1149.1-2013 TDR bit detail

Figure 5 shows scan path enabled when the INTEST3D instruction is loaded. The REG_1500 core is included in the path such that interconnect between the DIEWRAPPER and the REG_1500 wrapper can be performed without performing an instruction register scan.

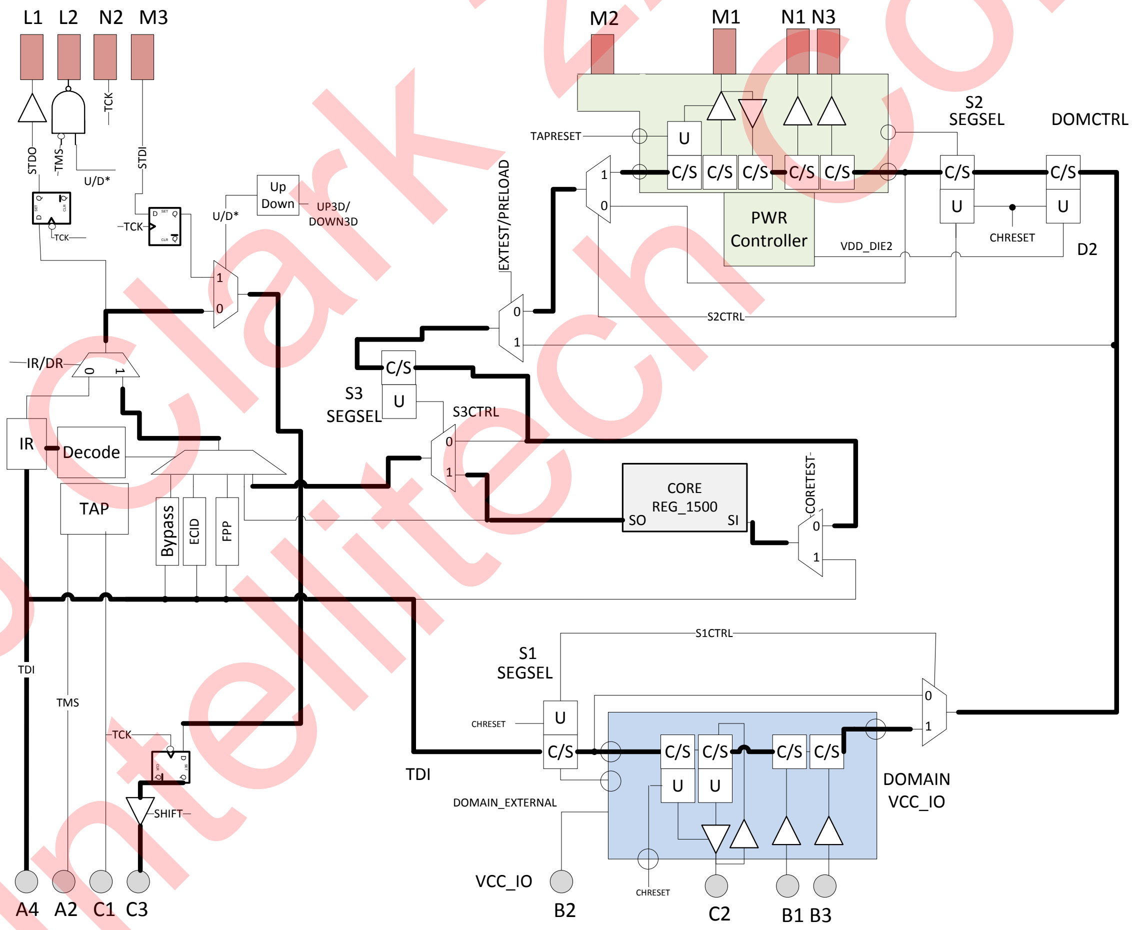


Figure 5. Bottom Die INTEST3D

Switching instructions may cause loss of TDR information depending on the implementation of the TDR. One possible approach using this architecture would be to perform WS_INTEST on the REG_1500 core using a 'short' scan-chain and CORETEST instruction (See Figure 6) and then followed by interconnect test patterns between REG_1500 core and the DIEWRAPPER register which presumably is a longer TDR. Another approach would be to have different target TDRs for EXTEST3D and INTEST3D, switching the REG_1500 segment via an instruction control of the mux. This is less favorable since the instruction decode needs to be routed across the die to the mux and it creates a need for two separate TDRs descriptions (DIEWRAPPERO and DIEWRAPPERI).

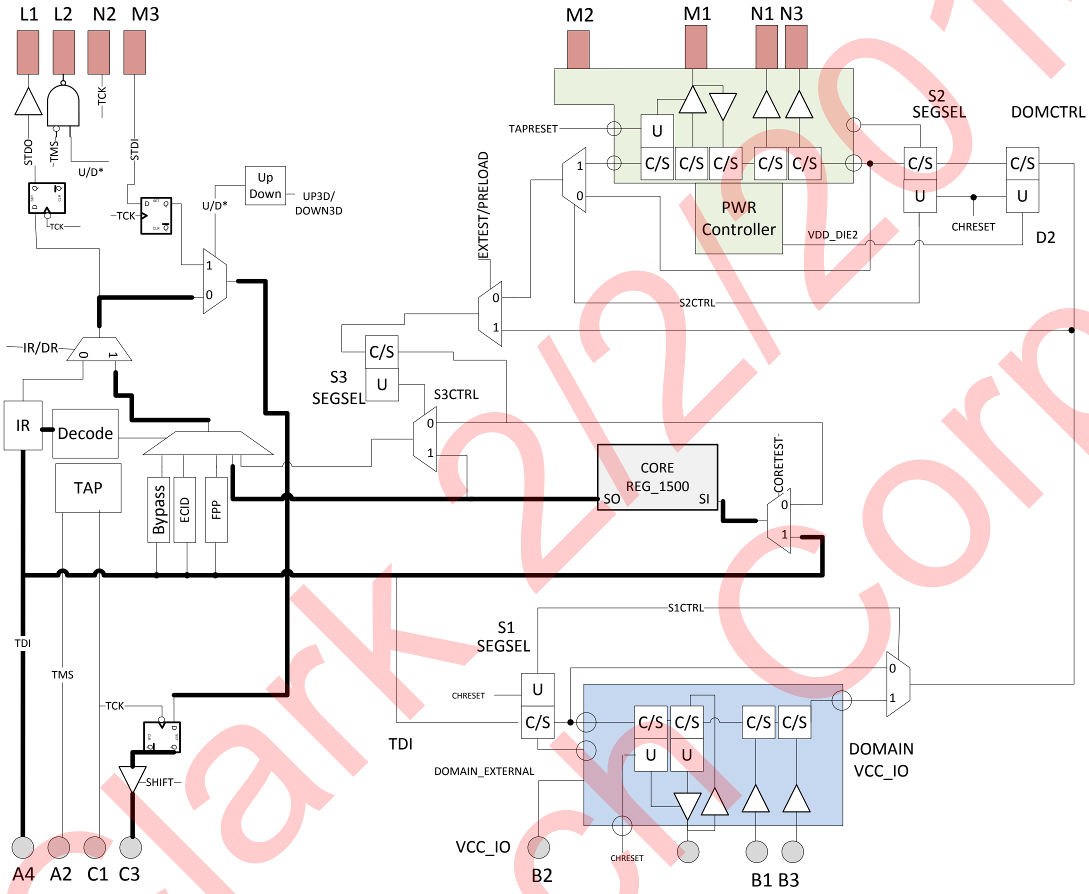


Figure 6. IEEE 1500 Core Test

When preparing to perform bottom die to middle die interconnect testing, the EXTEST3D instruction is loaded (inward and outward facing mode control signals are not shown). The path shown includes just the power domain controlled by DOMCTRL D2 and SEGSEL S1 which removes the board level boundary register portion from the scan chain and the REG_1500 core. In the example implementation, D2 enables power controller for the VDD_DIE2. The D2 control communicates with the PWR controller to enable power for the segment and for the middle die I/O above. This might be done in order to allow the two sections to be powered at the same time and hence not need power-domain crossing level shifters and isolation cells which take up area at the die to die interface. Dedicated control for power-controllers via scan chain control is a new capability of IEEE 1149.1-2013. The BSDL descriptions enable tools to understand the association of a scan-chain segment and its need to be powered prior to being enabled for scan. The power can be turned on via D2 and the state of the power domain can be checked by the capture value of SEGSEL S2.

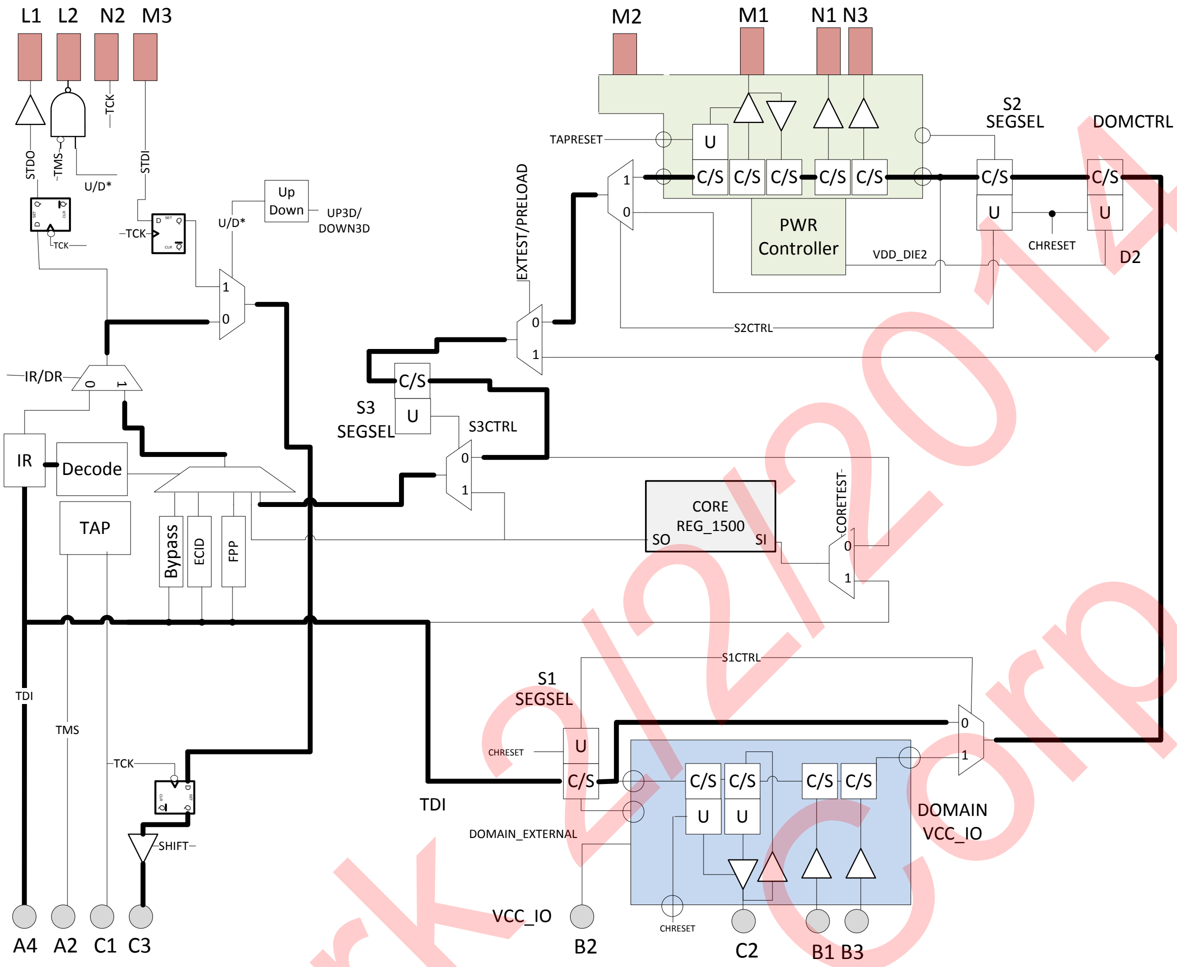


Figure 7. Bottom_Die in EXTEST3D

The proposed architecture with IEEE 1500 wrapped cores in switchable segments offers flexibility on whether the core is doubly wrapped (two muxes in the path) or not. In Figure 7, the assumption is that the DIEWRAPPER provides the entire interface to the TSVs/ubumps. However since the CORE segment can be put in the active scan path, then it is possible to make trade-offs between additional timing closure burdens on the designer and final test time.

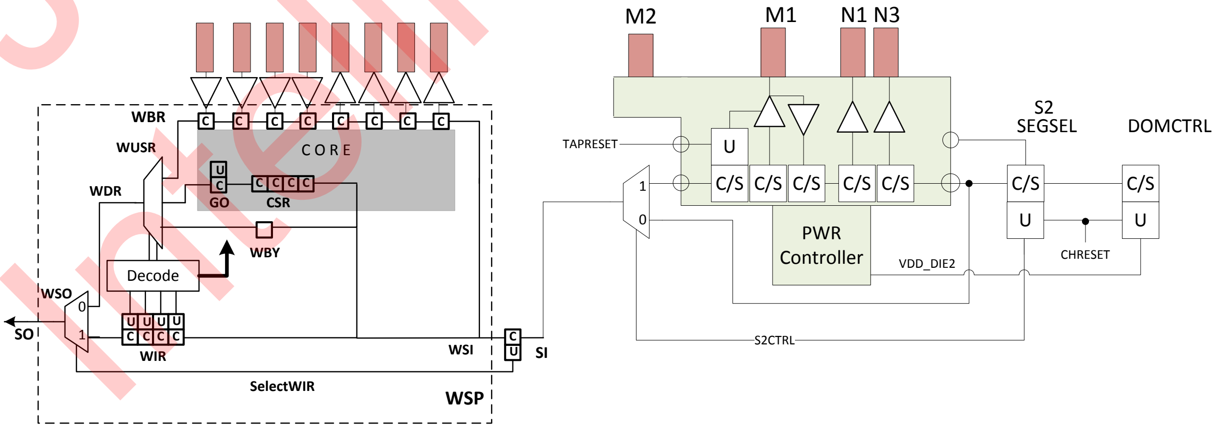


Figure 8. Core wrapper and die-wrapper in series

Figure 8 shows the details of the CORE if it had direct connections to the TSVs on the die. The core would be in series with Segment 2. This approach avoids the need for double wrapping but impacts test time as the entire core wrapper would need to be shifted through when only a few I/O of the core may actually go to TSVs. The P1838 standard doesn't need to specify rules on this as both approaches will work just one requiring more shift time but less work in meeting timing objectives. The timing is usually a solvable problem; hence what may be recommended is a dedicated DIEWRAPPER with some double-wrapping as necessary. A reasonable trade off would also be to enable FPP (Flexible Parallel Port) access to DIEWRAPPER and CORE WBR for faster production test. However, P1838 should offer guidance through a set of rules that for life-cycle testing the full interconnect test between die can be accomplished through the TAP albeit with longer shift times.

The figures use a new state machine called the Up/Down controller (See Figure 7). It is similar to the Test Mode Persistence Controller of IEEE 1149.1-2013. All die power up in the DOWN state via on-chip Power-on-Reset (similar to a TAP POR) or via TRST, if one is used by the die. If a TRST is not provided then a BYPASS-Escape mechanism enables a guaranteed way to exit from the UP state and back to a collapsed state. A tester would supply a logic "1" on TDI and enter the SHIFTIR state, shifting for a desirable number of clocks such that any die will revert back to the DOWN state.

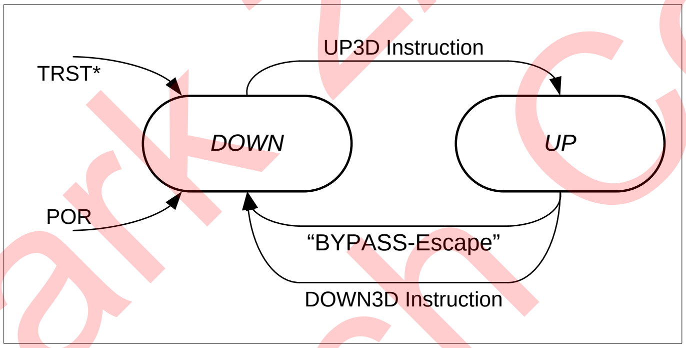
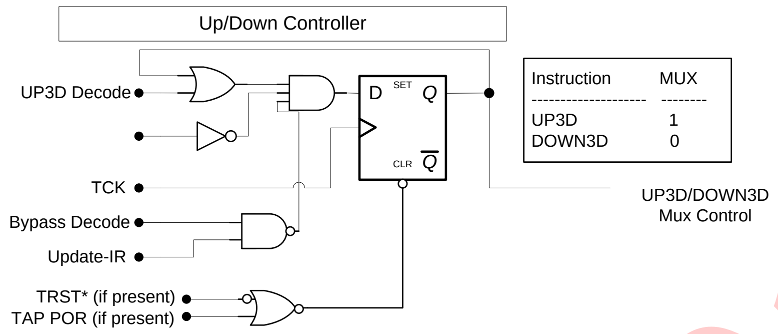


Figure 9. UP/DOWN State Machine

This is particularly useful when after opening the STDI-STDO path an open is found to the next die in the chain. It is also a required function for embedded system test where the die stack may be used in a highly reliable system in Avionics, Telecom or Medical when TRST is not present. Once powered-up, a single IR Scan can enable access to the next die without needing to do a DR scan. As shown in Figure 6, the die may be prepared for interconnect test by pre-loading the DIEWRAPPERO. Using an UP/DOWN controller which does not need a DR Scan enables the flexibility of pre-setup of a lower die prior to getting access to the next level die without incorporating a penalty in shifting through the long DIEWRAPPERO scan chain. The details of the Up/Down controller is shown in Figure 8. Note that the design uses the UPDATEIR state signal such that the Bypass instruction must be shifted and pass through the UPDATEIR state to clear the flop. Simply entering the Test-Logic-Reset state, which may load bypass does not clear the UP state. The controller is persistent from one instruction to another until DOWN3D or BYPASS. Care should be taken when designing P1838 die when a DEVICE_ID register is not implemented. If it is desired to have a short scan-chain capability without collapsing the die stack scan chain, an alternate instruction decode should be supplied which enables a 1 bit register in the DR scan path. This is illustrated in the supplied BSDL file.



Instruction	MUX
UP3D	1
DOWN3D	0

Figure 10. UP/DOWN Controller Schematic

The bottom die is prepared for die-to-die interconnect by loading the IR with the UP3D instruction. The enabled path is shown in Figure 9. In this mode the STDI/STDO is enabled allowing access to the middle die.

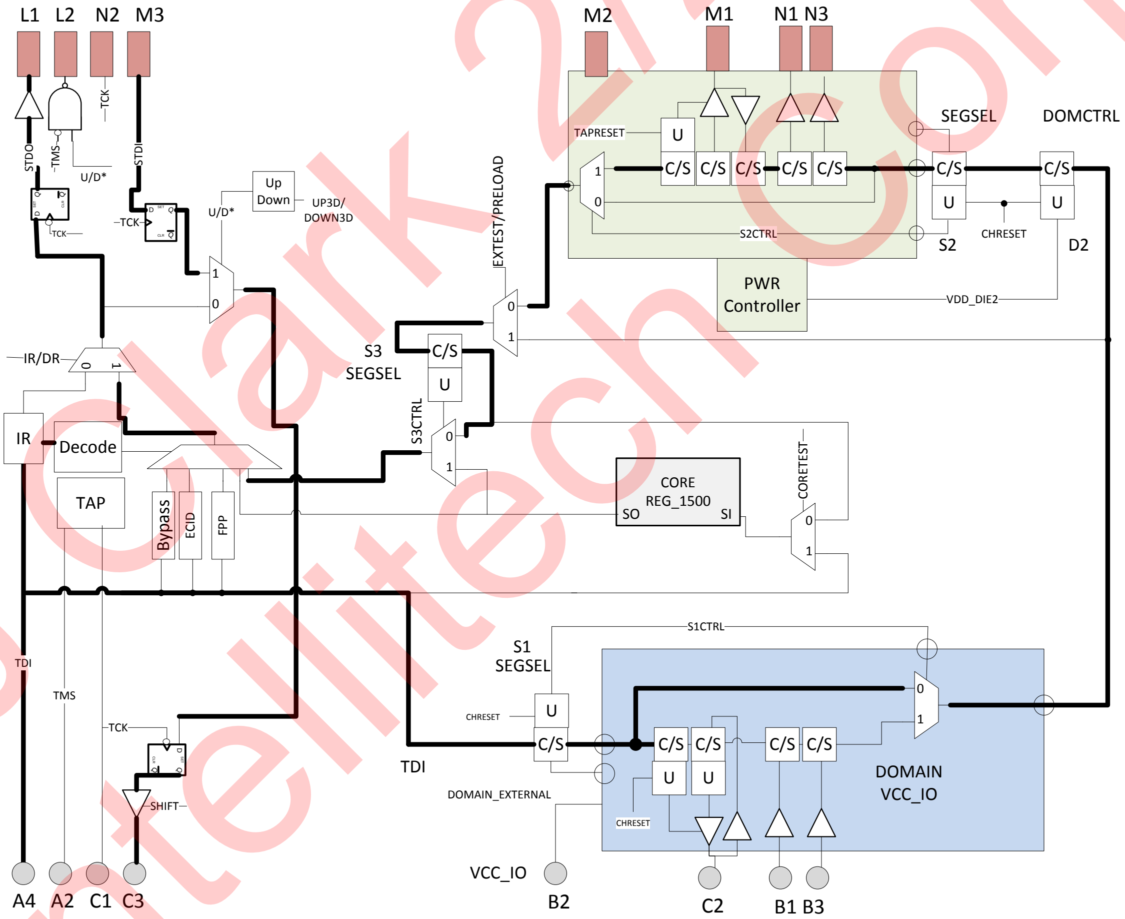


Figure 11. Preparing for die-to-die interconnect

BSDL Description

The highlighted yellow areas are the 'new' items needed for IEEE P1838. This includes a IEEE P1838 package description, five new Instructions and pipeline bit definitions. The rest is standard IEEE 1149.1. Only an incremental change is needed to support the unique aspects of a 3D stack.

```
entity Bottom_Die IS

-- Generic Parameter

generic (PHYSICAL_PIN_MAP : string := "DIE" );

-- Logical Port Description

port (

      GND: POWER_0 bit_vector (1 to 53);
      IO1: inout bit;
      IPAD1: in bit;
      IPAD2: in bit;
      TCK: in bit;
      TDI: in bit;
      TDO: out bit;
      TMS: in bit;
      VCCAUX: POWER_POS bit_vector (1 to 10);
      VCCINT: POWER_POS bit_vector (1 to 15);
      VCCO_0: POWER_POS bit_vector (1 to 6);
      VCCO_1: POWER_POS bit_vector (1 to 6);
      VCCO_2: POWER_POS bit_vector (1 to 6);
      VCC_IO: POWER_POS bit
); --end port list

-- Use Statements

use STD_1149_1_2013.all;
use STD_1838_201X.all;
use REG_1500.all;

-- Component Conformance Statement(s)

attribute COMPONENT_CONFORMANCE of Bottom_Die : entity IS
      "STD_1149_1_2013";

-- Device Package Pin Mappings

attribute PIN_MAP of Bottom_Die : entity IS PHYSICAL_PIN_MAP;

constant DIE: PIN_MAP_STRING:=
      "GND:(A1,A22,B7,B12,B16,C3,C20,D9,D14,F6," &
            "F11,F17,G2,G21,J4,J9,J11,J13,J14,J19," &
            "K10,K12,L2,L6,L9,L11,L13,L17,M10,M12," &
            "M14,M21,N11,N13,P4,P9,P10,P14,P19,T2," &
```

```
           "T12,T21,U6,U17,W10,W14,Y3,Y20,AA7,AA11," &
           "AA16,AB1,AB22)," &
     "IO1:C2," &
     "IPAD1:B1," &
     "IPAD2:B3," &
     "TCK:C1," &
     "TDI:A4," &
     "TDO:C3," &
     "TMS:A2," &
     "VCCAUX:(D12,E5,E18,H11,L4,M19,P11,V5,V18,W11)," &
     "VCCINT:(J10,J12,K9,K11,K13,L10,L12,L14,M9,M11," &
           "M13,N10,N12,N14,P13)," &
     "VCCO_0:(B5,B10,B14,B18,F9,F14)," &
     "VCCO_1:(E21,J17,K21,P17,P21,V21)," &
     "VCCO_2:(U9,U14,AA5,AA9,AA13,AA18)," &
     "VCC_IO:(B2)";


-- Scan Port Identification

attribute TAP_SCAN_OUT   of TDO : signal IS true;
attribute TAP_SCAN_IN    of TDI : signal IS true;
attribute TAP_SCAN_CLOCK of TCK : signal IS (50.0e6, both);
attribute TAP_SCAN_MODE  of TMS : signal IS true;


-- Instruction REGISTER Description

attribute INSTRUCTION_LENGTH of bottom_die : entity IS 8;

attribute INSTRUCTION_OPCODE of bottom_die : entity IS

   "EXTEST      (00001111)," &
   "SAMPLE      (00000001)," &
   "PRELOAD     (00000001)," & -- Same as SAMPLE
   "ECIDCODE    (00000011)," & -- perhaps required for P1838?
   "HIGHZ       (00001010)," &
   "CLAMP_HOLD       (00110010)," &
   "CLAMP_RELEASE    (00110011)," &
   "CORETEST    (00001001)," &
-- 3D instructions
   "INTEST3D    (00000010)," &
   "EXTEST3D    (00100010)," &
   "UP3D        (00100011)," &  -- sticky function like CLAMP_HOLD
   "DOWN3D      (00100111)," &  -- sticky function like CLAMP_HOLD
   "FPPCNTRL    (00101000)," &
   "BYPASSALT   (11000000)," &
   "BYPASS      (11111111, 00000000)";

attribute INSTRUCTION_CAPTURE of bottom_Die : entity IS "XXXXXX01";

-- REGISTER Access Description

attribute REGISTER_ACCESS of bottom_die : entity IS
"DIEWRAPPER[*] (EXTEST3D, INTEST3D)," &   -- length can be deferred
"FPP[4]        (FPPCNTRL)," &
"BYPASS        (HIGHZ, BYPASS, BYPASSALT, UP3D, DOWN3D)," &
```

```
"ECID[32]      (ECIDCODE)," &
-- boundary + short example REGISTER - not real world
"REG_1500[*]   (CORETEST)," &   -- length can be defferred
"TMP_STATUS ( CLAMP_HOLD, CLAMP_RELEASE ),"& -- length described
"BOUNDARY  (EXTEST, SAMPLE, PRELOAD )";

-- Boundary-Scan REGISTER Description

attribute BOUNDARY_LENGTH of Bottom_Die : entity IS 4;

attribute BOUNDARY_SEGMENT of Bottom_Die : entity IS
"priboundary [4] ("&
-- num cell port function safe [ccell dISval rslt]
     "   0 (BC_2, IPAD2, input, X)";
     "   1 (BC_1, IPAD1, input, X)," &
     "   2 (BC_7 , IO1, bidir, X, 3, 1, PULL1)," & --  PAD100
    "   3 (BC_2, *, controlr, 1)" &
 " ) ";

attribute REGISTER_ASSEMBLY of Bottom_Die : entity IS
"boundary ( "&
"(S1 IS SegSel DOMAN_EXTERNAL(VCC_IO) Segment(S1CTRL) TAPReset), "&
"(Seg1 IS priboundary), "&
"(Pri_mux IS SegMux Segment(S1CTRL)), "&
" )" ;


-- you can describe DIEWRAPPER here or describe in a package file


attribute REGISTER_ASSEMBLY of bottom_die : entity IS
"REG_1500 ( " &    -- define REGISTER for CORETEST instruction
    "(CORE1 IS REG_1500 )  "&
-- instantiate instance of REG_1500 description from package file
  " )" ;


attribute REGISTER_ASSEMBLY of bottom_die : entity IS

"PRIMARYSEG ( " &   -- define REGISTER for primary side
    "(PRIREG [4] )  "&   -- other attributes possible
    " ), "&
"SECONDARYSEG ( " &   -- define REGISTER for primary side
    "(OUTEN   [1] SHARED CHRESET ),  "&   -- other attributes possible
    "(SECBIDIRO[1] SHARED NOUPD )  "&
    "(SECBIDIRI[1] SHARED NOUPD )  "&
    "(SECREG   [2] SHARED NOUPD )  "&
    " ), "&
"DIEWRAPPER( "&
-- SI
-- blocked CLAMP_HOLD Reset
"(S1 IS SegSel Segment(S1CTRL) DOMAIN_EXTERNAL(VCC_IO) CHReset), "&
"(Seg1 IS PrimarySeg), "&
"(Pri_mux IS SegMux Segment(S1CTRL)), "&
-- blocked CLAMP_HOLD Reset
"(D1 IS DomCtrl DOMAIN(VDD_DIE2) CHReset), "&
"(S2 IS SegSel DOMAIN(VDD_DIE2) Segment(S2CTRL) CHReset), "&
"(Seg2 IS SecondarySeg), "&
"(Sec_mux IS SegMux Segment(SEC) ), "&
"(S2 IS SegSel Segment(S3CTRL) CHReset), "&
"(CORE1 IS REG_1500), "&
"(S3_mux IS SegMux Segment(S3CTRL) ) "&
") ";
-- SO



attribute REGISTER_ASSOCIATION of bottom_die : entity IS
"S1 : port(VCC_IO) ";

attribute REGISTER_ASSOCIATION of bottom_die : entity IS
"Seg2 : user TSVMAP (N3, N1, M1) ";


attribute TDI_PIPELINE of bottom_die : entity IS "0";
attribute TDO_PIPELINE of bottom_die : entity IS "1";

end Bottom_Die;
```

The BSDL introduces two new attributes from the P1838 standard package file, TDI_PIPELINE and TDO_PIPELINE which describes the number of pipeline flops present on the TDO and TDI side of the TAP. This should be very easy for current emulation tools or FPGA tools to adopt to move from 2D to a 3D context.

Also of note is the mapping of the TSVs to physical locations enabled by the REGISTER_ASSOCIATION. Since it is possible to write a simple interconnect test between die in IEEE 1149.1-2013 PDL then such information is useful in identifying the physical locations where registers are failing.

A middle die is illustrated in Figure 10. Since the primary interface of the middle die is not controlled by an IEEE 1149.1 boundary scan register the design can be more flexible and similar to a IEEE 1500 wrapper. Note that in DOMAIN PRI_VDD, an update register is not present on C2 as it is not required. However, controls going to a bidirectional should include the update register in order to hold a safe state and avoid die to die contentions.

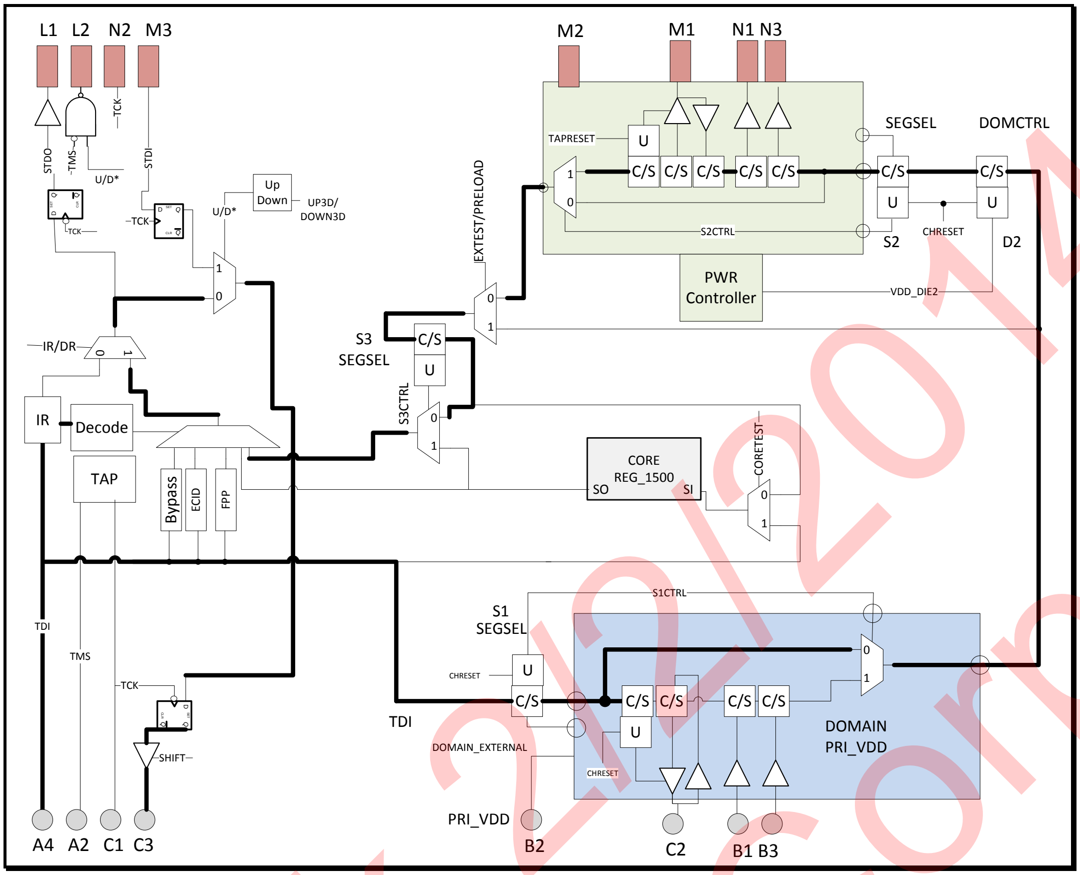


Figure 12. Middle Die in EXTEST3D

Note the S1 SEGSEL will check to see if the segment in DOMAIN PRI_VDD is "ready to scan". It's possible that the TAP is in a 'always powered' domain and the die is segmented. In this case PRI_VDD is powered by the lower die so the interface between bottom and middle die can be powered at the same time and therefore not needing isolation, retention or level shifters designed at the TSV boundary.

The middle die is also IEEE 1149.1-2013 compliant in order to leverage end user's large ecosystem of supporting tools. There is no boundary register since there are no system inputs and outputs going to the middle die. In EXTEST mode, the bypass register acts as a 1 bit 'internal' boundary register the minimum requirement. (See BSDL description).

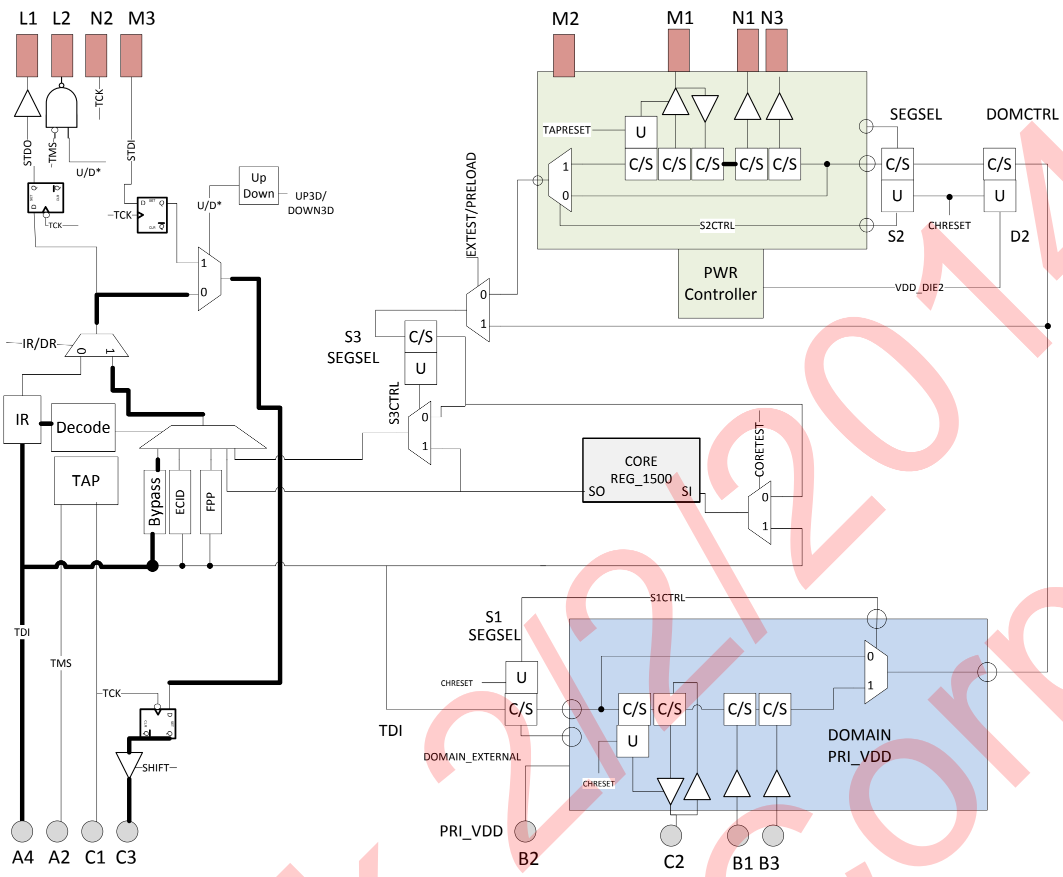


Figure 13.   Middle-Die in EXTEST (not EXTEST3D)

```
entity Middle_Die IS

-- Generic Parameter

generic (PHYSICAL_PIN_MAP : string := "DIE" );

-- Logical Port Description

port (

     GND: POWER_0 bit_vector (1 to 53);
     IO1: inout bit;
     IPAD1: in bit;
     IPAD2: in bit;
     TCK: in bit;
     TDI: in bit;
     TDO: out bit;
     TMS: in bit;
     VCCAUX: POWER_POS bit_vector (1 to 10);
     VCCINT: POWER_POS bit_vector (1 to 15);
     VCCO_0: POWER_POS bit_vector (1 to 6);
     VCCO_1: POWER_POS bit_vector (1 to 6);
     VCCO_2: POWER_POS bit_vector (1 to 6);
     PRI_VDD: POWER_POS bit
); --end port list

-- Use Statements

use STD_1149_1_2013.all;
use STD_1838_201X.all;   -- the year has not been determined
use REG_1500.all;

-- Component Conformance Statement(s)

attribute COMPONENT_CONFORMANCE of Middle_Die : entity IS
      "STD_1149_1_2013";

-- Device Package Pin Mappings

attribute PIN_MAP of Middle_Die : entity IS PHYSICAL_PIN_MAP;

constant DIE: PIN_MAP_STRING:=
      "GND:(A1,A22,B7,B12,B16,C3,C20,D9,D14,F6," &
           "F11,F17,G2,G21,J4,J9,J11,J13,J14,J19," &
           "K10,K12,L2,L6,L9,L11,L13,L17,M10,M12," &
           "M14,M21,N11,N13,P4,P9,P10,P14,P19,T2," &
           "T12,T21,U6,U17,W10,W14,Y3,Y20,AA7,AA11," &
           "AA16,AB1,AB22)," &
      "IO1:C2," &
      "IPAD1:B1," &
      "IPAD2:B3," &
      "TCK:C1," &
      "TDI:A4," &
      "TDO:C3," &
      "TMS:A2," &
      "VCCAUX:(D12,E5,E18,H11,L4,M19,P11,V5,V18,W11)," &
      "VCCINT:(J10,J12,K9,K11,K13,L10,L12,L14,M9,M11," &
           "M13,N10,N12,N14,P13)," &
      "VCCO_0:(B5,B10,B14,B18,F9,F14)," &
      "VCCO_1:(E21,J17,K21,P17,P21,V21)," &
      "VCCO_2:(U9,U14,AA5,AA9,AA13,AA18)," &
      "PRI_VDD:(B2)";


-- Scan Port Identification

attribute TAP_SCAN_OUT   of TDO : signal IS true;
attribute TAP_SCAN_IN    of TDI : signal IS true;
attribute TAP_SCAN_CLOCK of TCK : signal IS (50.0e6, both);
attribute TAP_SCAN_MODE  of TMS : signal IS true;


-- Instruction REGISTER Description

attribute INSTRUCTION_LENGTH of Middle_Die : entity IS 8;

attribute INSTRUCTION_OPCODE of Middle_Die : entity IS

   "EXTEST      (00001111)," &
   "SAMPLE      (00000001)," &
   "PRELOAD     (00000001)," & -- Same as SAMPLE
   "ECIDCODE    (00000011)," & -- perhaps required for P1838?
   "HIGHZ       (00001010)," &
   "CLAMP_HOLD       (00110010)," &
   "CLAMP_RELEASE    (00110011)," &
   "CORETEST    (00001001)," &
-- 3D instructions
   "INTEST3D    (00000010)," &
   "EXTEST3D    (00100010)," &
   "UP3D        (00100011)," &  -- sticky function like CLAMP_HOLD
   "DOWN3D      (00100111)," &  -- sticky function like CLAMP_HOLD
   "FPPCNTRL    (00101000)," &
   "BYPASSALT   (11000000)," &
   "BYPASS      (11111111, 00000000)";

attribute INSTRUCTION_CAPTURE of Middle_Die : entity IS "XXXXXX01";

-- REGISTER Access Description

attribute REGISTER_ACCESS of Middle_Die : entity IS
"DIEWRAPPER[*] (EXTEST3D, INTEST3D)," &   -- length can be deferred
"FPP[4]        (FPPCNTRL)," &
"BYPASS        (HIGHZ, BYPASS, BYPASSALT, UP3D, DOWN3D)," &
"ECID[32]      (ECIDCODE)," &
-- boundary + short example REGISTER - not real world
"REG_1500[*]   (CORETEST)," &   -- length can be defferred
"TMP_STATUS ( CLAMP_HOLD, CLAMP_RELEASE ),"& -- length described
"BOUNDARY  (EXTEST, SAMPLE, PRELOAD )";

-- Boundary-Scan REGISTER Description

attribute BOUNDARY_LENGTH of Middle_Die : entity IS 1;

attribute BOUNDARY_REGISTER of Middle_Die : entity IS
-- num cell port function safe [ccell dISval rslt]
     "   0 (BC_1, *, internal, X)";   -- this is the bypass register
" ) ";


-- you can describe DIEWRAPPER here or describe in a package file

attribute REGISTER_ASSEMBLY of Middle_Die : entity IS
"REG_1500 ( " &   -- define REGISTER for CORETEST instruction
    "(C1 IS REG_1500 )  "&
-- instantiate instance of REG_1500 description from package file
  " )" ;


attribute REGISTER_ASSEMBLY of Middle_Die : entity IS

"PRIMARYSEG ( " &   -- define REGISTER for primary side
    "(PRIREG1 [1] SHARED CHRESET ),  "&   -- other attributes possible
    "(PRIREG2 [3] SHARED NOUPD)  "&   -- other attributes possible
    " ), "&
"SECONDARYSEG ( " &   -- define REGISTER for primary side
    "(OUTEN   [1] SHARED CHRESET ),  "&   -- other attributes possible
    "(SECBIDIRO[1] SHARED NOUPD )  "&
    "(SECBIDIRI[1] SHARED NOUPD )  "&
    "(SECREG  [2] SHARED NOUPD )  "&
    " ), "&
"DIEWRAPPER( "&
-- SI
-- blocked CLAMP_HOLD Reset
"(S1 IS SegSel Segment(S1CTRL) DOMAIN_EXTERNAL(PRI_VDD) CHReset), "&
"(Seg1 IS PrimarySeg), "&
"(Pri_mux IS SegMux Segment(S1CTRL)), "&
-- blocked CLAMP_HOLD Reset
"(D1 IS DomCtrl DOMAIN(VDD_DIE2) CHReset), "&
"(S2 IS SegSel DOMAIN(VDD_DIE2) Segment(S2CTRL) CHReset), "&
"(Seg2 IS SecondarySeg), "&
"(Sec_mux IS SegMux Segment(SEC) ), "&
"(S2 IS SegSel Segment(S3CTRL) CHReset), "&
"(C1 IS REG_1500), "&
"(S3_mux IS SegMux Segment(S3CTRL) ) "&
") ";
-- SO


attribute REGISTER_ASSOCIATION of Middle_Die : entity IS
"S1 : port(PRI_VDD) ";

attribute REGISTER_ASSOCIATION of Middle_Die : entity IS
"Seg1 : user TSVMAP (C2, B1, B3), " &
"Seg2 : user TSVMAP (N3, N1, M1) ";


attribute TDI_PIPELINE of Middle_Die : entity IS "0";
attribute TDO_PIPELINE of Middle_Die : entity IS "1";

end Middle_Die;
```

The CORE which is being accessed can be described in a IEEE 1149.1-2013 package file.   An example description is included below.   A schematic representation is shown in Figure 12 with a package file description in 1149.1-2013 format.

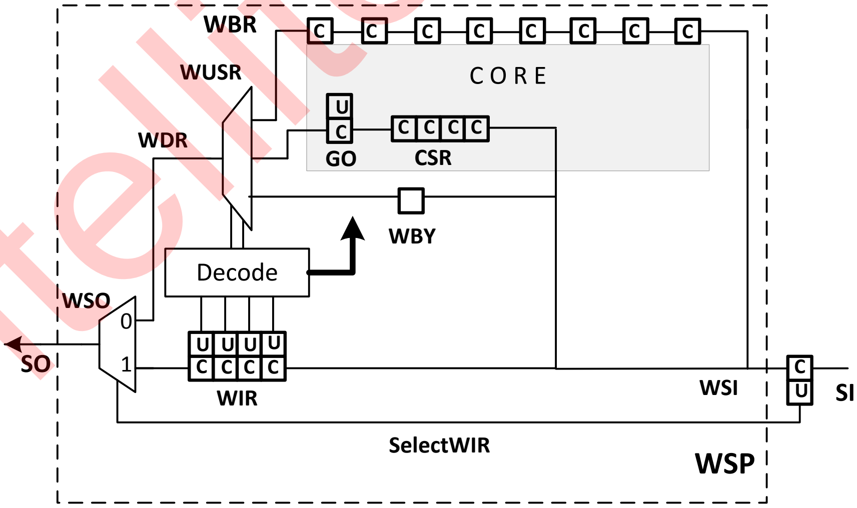


Figure 14. IEEE 1500 example architecture for CORE

BSDL Package File for wrapped core.

```
-- Supplied by MyCorp for REG_1500 version 1.0
package REG_1500 is
   use STD_1149_1_2013.all;
end REG_1500;

package body REG_1500 is

   use STD_1149_1_2013.all;

Attribute REGISTER_MNEMONICS of REG_1500  : package is
  "WIR_decode  ( "&
    "WS_BYPASS  (0B0000) <Wrapper Instruction>, "&
    "WS_EXTEST  (0B0001) <Wrapper External Boundary Instruction>, "&
    "WS_INTEST  (0B0010) <Wrapper Internal Boundary Instruction>, "&
    "WS_BIST    (0B0100) <BIST Instruction>, "&
    "WP_ALL     (0B1xxx) <Wrapper Parallel instructions> "&
    " )," &
  "BISTGROUP ( "&
    "Disable  (0B0) < BIST has not been enabled >, "&
    "Enable   (0B1) < BIST enabled > "&
    " ),"&
  "STATGROUP ( "&
    "PASS     (0B1001),  "&
    "FAIL     (0B0111)   "&
    " )," &
  "MODEGROUP ( "&
    "MODE0    (0X0),  "&
    "MODE3    (0X3)   "&
    " )";

Attribute REGISTER_ASSEMBLY of REG_1500  : package IS
  "REG_1500 ( " &  -- The Select WIR bit and the Wrapper Serial Port
    -- Reset to WBY
    "(SELWIR [1] DelayPO ResetVal(0b0) TAPReset ),  "&
    "(WSP IS WSP_MUX) "&
  " ), "&
  "WSP_MUX ( "&   -- The outer selectable segments: WIR and WDR
    "(SelectMUX "&
      -- Reset to WBY
      "(WIR IS WIR_Seg), "&
      "(WDR IS WDR_MUX) "&
      "SelectField (SELWIR) "&
      "SelectValues ((WIR : 0b1) (WDR : 0b0)) "&
    " ) "&
  " ), "&
  "WIR_Seg (  (WIR_field [4] DelayPO "&
              "ResetVal(WIR_decode(WS_BYPASS)) TAPReset )    ), "&
  "WDR_MUX ( "&  -- The inner selectable segments: WBY, WBR, and Wusr
    "(SelectMUX "&
      "(WBY  IS Reg_WBY CAPTURES(0) ), "&
      "(WBR  IS Reg_WBR), "&
      "(WUSR IS Reg_WUSER) "&
      "SelectField (WIR) "&
      "SelectValues ("&
     -- segment : selectfield value to access segment
        "(WBY  : WS_BYPASS, WP_ALL) "&
        "(WBR  : WS_EXTEST, WS_INTEST) "&
        "(WUSR : WS_BIST) "&
        " ) "&
    " ) "&
  " ), "&
  "REG_WBY   ( (WBY[1] NOPO)), " &
  "REG_WBR   ( (WBR[8] NOUPD SHARED )), " &
  "REG_WUSER ( (  CSR[4] CAPTURES(STATGROUP(-)) " &
                     "DEFAULT(MODEGROUP(MODE0)) NOUPD )," &
  "            (  GO [1] ResetVal(BISTGROUP(Disable)) TapReset )   )"
;

end REG_1500;
```

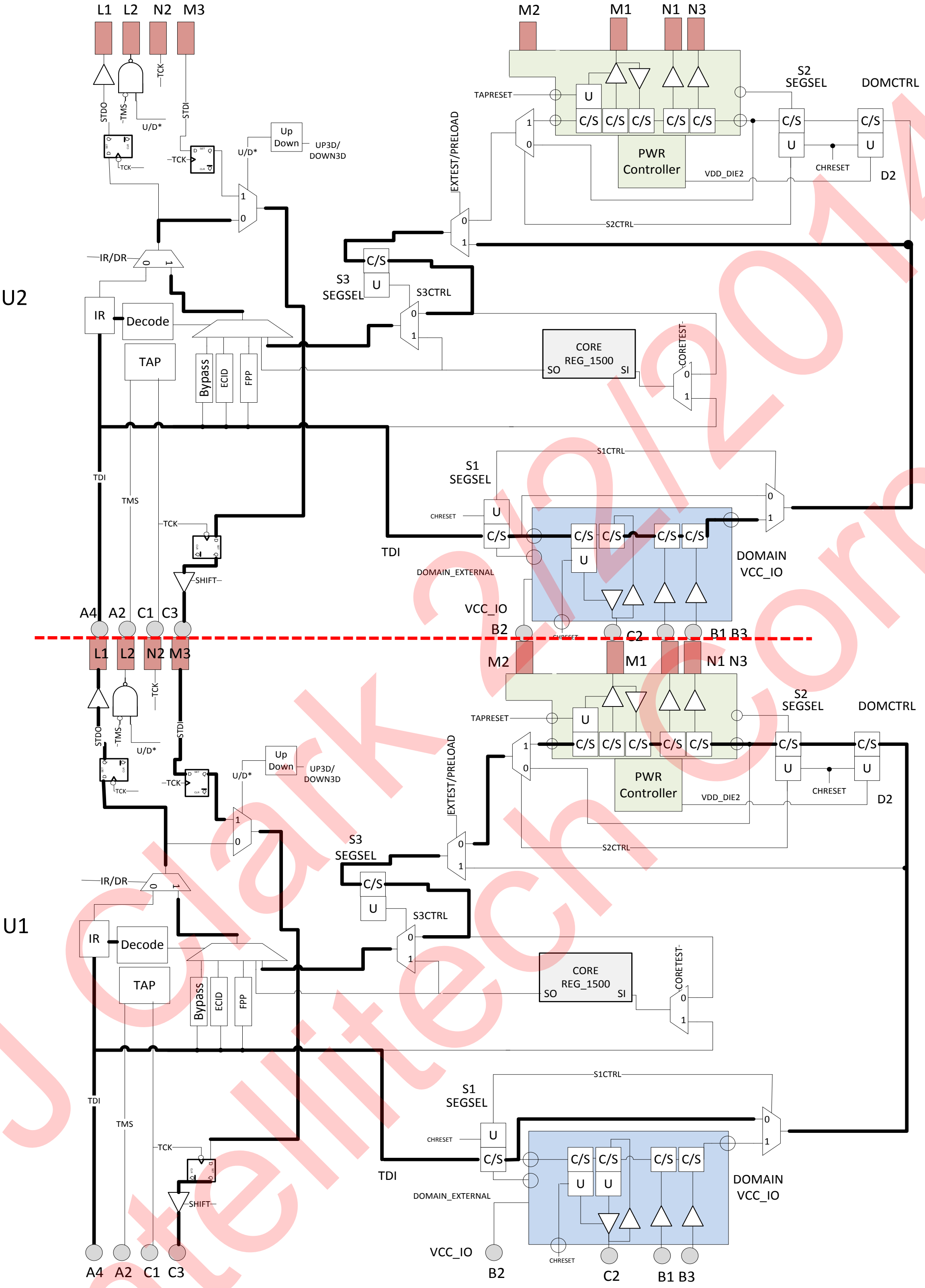


Figure 15. Two die stack

PDL Code

Die and stack independent PDL code can be provided as a re-usable set of procedures.

Reg_1500.pdl

```
# Supplied by MyCorp for REG_1500 version 1.0
iPDLLevel 0 -version STD_1149_1_2013
iProcGroup REG_1500
# check that bypass register can be scanned
iProc -export check_bypass { } {
iWrite WIR WS_BYPASS ;# Use WS_BYPASS and not WP_ALL
iRead WBY 0 ;# = WDR.WBY, only one WBY in package
iApply
}
# Set up mode and execute REG_1500 BIST in 1 scan operation
iProc -export start_bist { mode } {
# CSR is documented to be a capture and shift register only.
# GO has capture, shift, and update.
iWrite CSR $mode ;# = WDR.Wusr.CSR, only one CSR in package
iWrite GO Enable ;# = WDR.Wusr.GO, only one GO in package
iApply
iRunLoop 100000
}
# Check BIST results
iProc -export check_bist { instance mode } {
iRead CSR PASS ;# = WDR.Wusr.CSR, only one CSR in package
iApply -nofail
ifFalse
iSetFail "$instance REG_1500 BIST test with mode = $mode failed\n"
ifEnd
}
```

PDL can also be supplied by the die provider to be used in the stack.   The die provider can leverage the routines already developed for the IEEE 1500 CORE.

Bottom_Die

```
# Bottom Die PDL
iPDLLevel 0 -version STD_1149_1_2013
iSource REG_1500.PDL
iProcGroup BOTTOM_DIE

iProc main {
```

```
iCall CORE1.start_bist MODE0
iCall CORE1.check_bist CORE1 MODE0
}
<EOF>

# Middle Die PDL
iPDLLevel 0 -version STD_1149_1_2013
iSource REG_1500.PDL
iProcGroup MIDDLE_DIE

iProc main {

iCall C1.start_bist MODE0
iCall C1.check_bist C1 MODE0
}
<EOF>
```

```
# Stack level PDL
iPDLLevel 0 -version STD_1149_1_2013
iSource BOTTOM_DIE.PDL
iSource MIDDLE_DIE.PDL

iProc main {
iMerge -begin
iCall U1.main
iCall U2.main
iMerge -end
}

iProc -export interconnect {
# for next week.
#iWrite U1.Seg2.OUTEN
#iWrite U1.Seg2.SECBIDIR
#iWrite U1.Seg2.SECREG
}
<EOF>
```

Optional

It's possible to gate TCK going to the next die if desired. A TAP in reset normally has little switching and therefore very little current draw when not in use. The TCK can be gated with the Up/Down* signal, if desired. Using this approach obviously prevents the upper die from 'running' a TCK BIST while the stack is collapsed. This seems to have little real world utility as BIST is typically run using a System clock and not the TCK. When the stack is collapsed the TAPs go into test logic reset. It would be much more practical to leave the stack expanded and run BIST or whatever needs to be done. The penalty would be having a few bypass registers in the path while test continues during another die's BIST. Leaving this to the stack maker to implement as needed appears to be completely acceptable position for the P1838 standard to take.

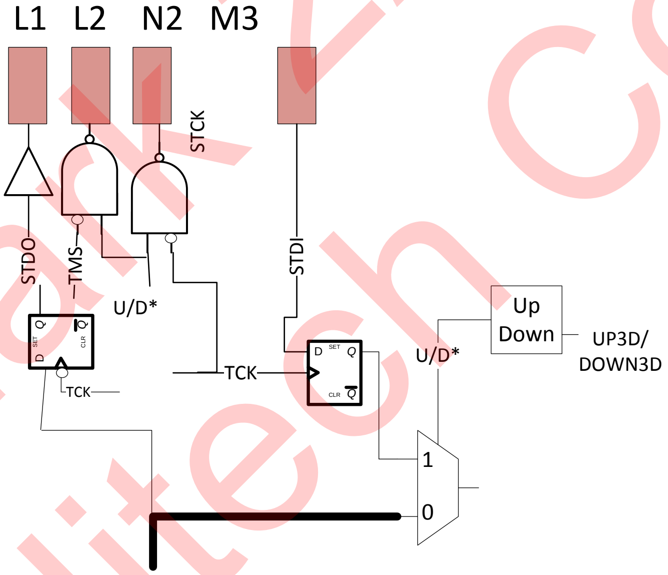


Figure 16. Optional TCK gating detail