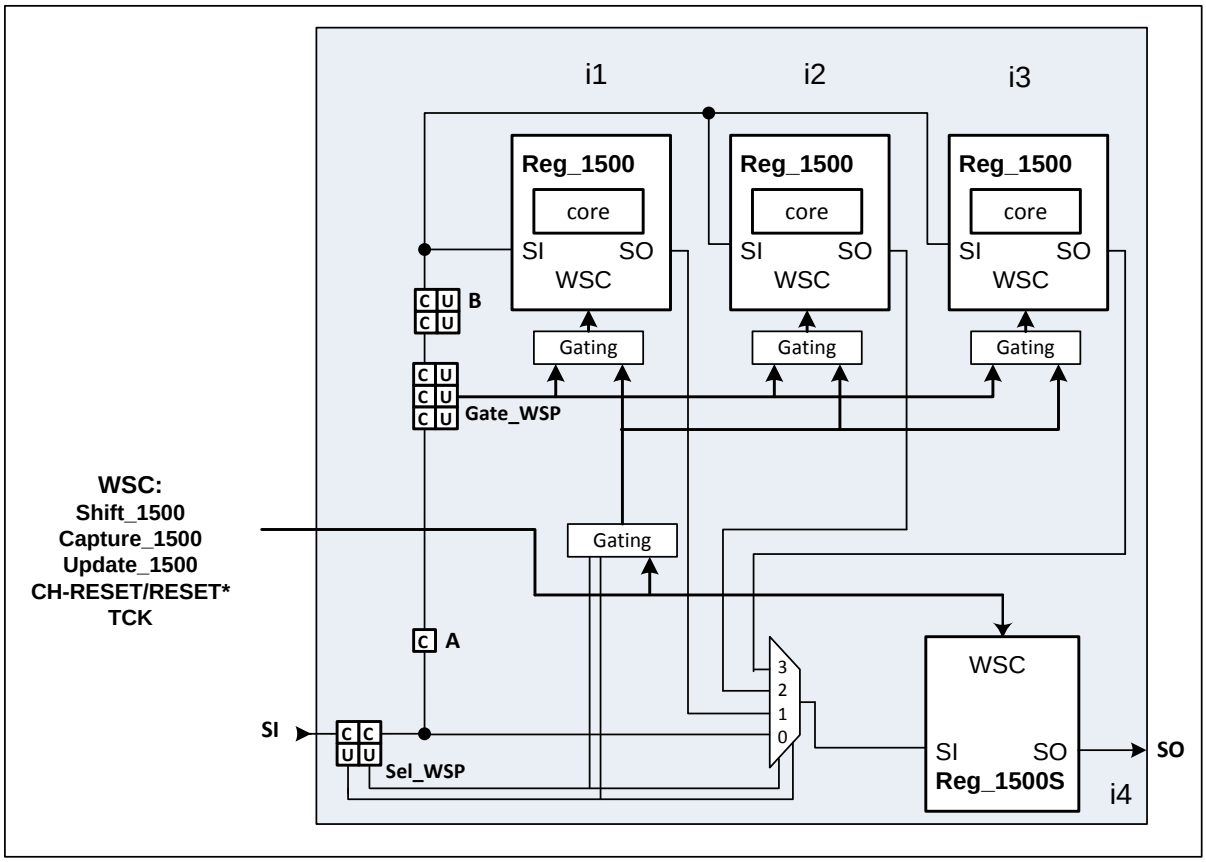
Fix for global CH-RESET with local resets

On IEEE 1500 architectures

CJ Clark

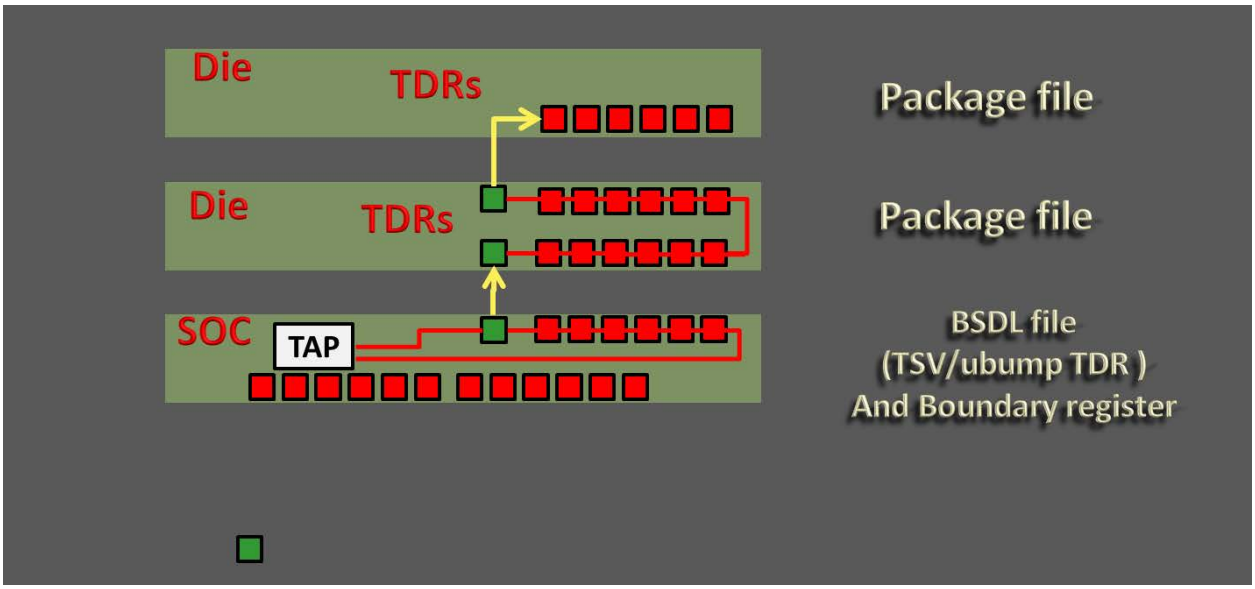
Version 1.2



Currently we do not allow flexible separation of reset. Entire IC or multiple die in an IC take a global RESET or CH-RESET. Blocking RESET on the init_data and boundary register is very desirable, however once that is done the same CH-RESET goes to all internal BIST and BISRs, all 1500 signals. There is no granularity. What we're asking the IC community to do is broken since once one uses the benefits of CLAMP_HOLD to block resets for the boundary, then all resets inside the IC are blocked.

HIERRESET is one approach. HIERRESET requires the circuitry to be in a selectable segment and then it requires that when the segment is collapsed the circuits are reset. This affects test scheduling, meaning I may do something above in i1, i2, i3, i4 and kick off a BIST which requires many clock cycles. But I'm forced to shift through the entire architecture during testing of surrounding sections of the IC because if I collapse it, all the BIST engines stop. HIERRESET is nice but it is not as flexible as LOCALRESET due to this problem.

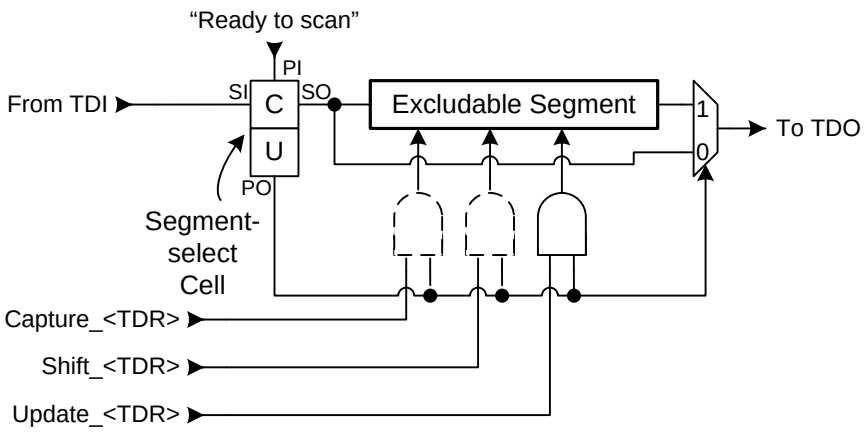
The standard is '3D SIC' ready. One can describe a bottom die with a TAP and then have a package file for each DIE which is included.



If IEEE 1500 paths are used to connect each die, then just one single RESET is passed to all die. If this is TMSRESET or CHRESET then the entire stack is responding to the reset. Testing which is occurring would be interrupted. With local reset then 'blocking' reset will work but not be 'global' in nature.

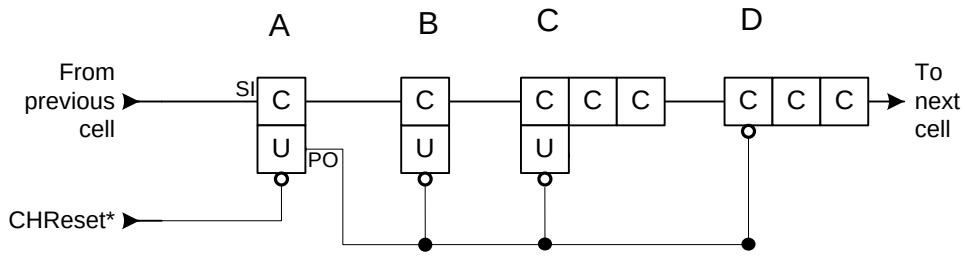
One way to fix this is to have multiple bits in the TMP_Status register. Another way is to allow local resets. A scan cell with all the rules of a SEGSEL could act as the reset for various TDR and WIRs inside the IC.

Local Reset is just a more generic case of hierarchical reset and is not associated with a SEGMENT or need a segment to collapse. The cell itself is the same as a SEGSEL has to POR up with a 0.



SEGSEL

Bit A below is a Local Reset. It takes one of the RESETs, preferably CHRESET. It's PO then is used to reset bits B, C and D.



If a bit is specified as NOUPD then the reset goes to the c/s flop. Note that the output of the LocalReset is very much the same as the output of a SEGSEL, instead it does not go to a MUX but the asynchronous set/clear of a Flop.

We have this:

<reset assignment>::= **PORRESET** | **TRSTRESET** | **TAPRESET** | **CHRESET** | **DOMPOR** | **HIERRESET**

<domain assignment>::= <association type> <left paren> <association name> <right paren>
<association type>::= **DOMAIN** | **DOMAIN_EXTERNAL** | **SEGMENT**
<association name>::= **Error! Reference source not found.**

I'm just looking for a way to describe resets which come from register bits. They are allowed alreadybut the problem is the tool doesn't know it's there. The Global CH-RESET isn't useful as while its holding reset for the init_data and boundary register, it is being distributed to various parts of the IC preventing one from resetting IP blocks as needed.

The description which may fix this, rather than changing TMP_STATUS is:

<reset assignment>::= **PORRESET** | **TRSTRESET** | **TAPRESET** | **CHRESET** |
DOMPOR | **HIERRESET** | <local reset assignment>

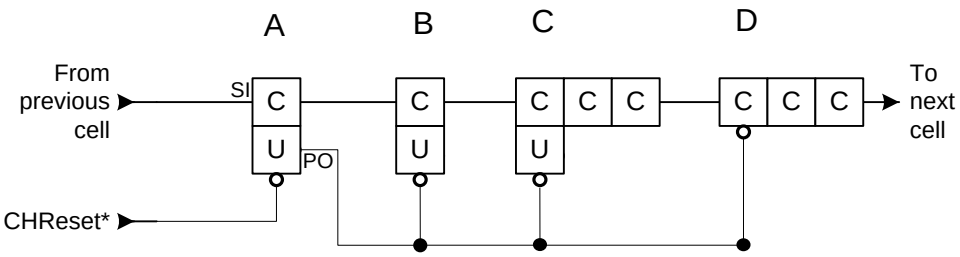
<local reset assignment>::= < reset type> <left paren> <reset name> <right paren>
<reset type>::= **RESETOUT** | **RESETINPUT**
<reset name>::= **Error! Reference source not found.**

One and only one RESETOUT (reset out) of the same <reset name> would be allowed and as many RESETINPUT (reset in) of the same name would be allowed. More than one <reset assignment> could be used, such as PORRESET, but these would be redundant, since the TDR bits would be reset at power up via the RESETOUT cell. Similarly, combining TAPRESET with a RESETINPUT for a local reset, effectively breaks the advantage of having the local reset.

A <reset name> which does not have a <reset type> of RESETOUT is an error (no source to drive the reset)

This would tell tools the relationship of the reset. Where it is generated and where it is distributed in the design.

```
  attribute REGISTER_FIELDS of Reg_Types : package is
    "Regs [8] ( "&
      "(needsreset[1] IS (7) RESETINPUT(rst1)  ) "&
      "(LocalReset [1] IS (6) RESETOUT(rst1) PULSE0 PORRESET ), "&
      "(ReadWrite [1] IS (5)), "&           -- Fig Error! Reference source
not found.
      "(WriteOnly [1] IS (4) NoPI), "&      -- Fig Error! Reference source
not found.
      "(ReadOnly  [1] IS (3) NoPO), "&      -- Fig Error! Reference source
not found.
      "(ShiftOnly [1] IS (2) NoPI NoPO), "&  -- Fig Error! Reference source
not found.
      "(SelfMon   [1] IS (1) MON), "&       -- Fig Error! Reference source
not found.
      "(Pulse1Mon [1] IS (0) PULSE1 MON) "& -- Fig Error! Reference source
not found.
      "      )";
```




```
attribute REGISTER_FIELDS of MyReg : package is
  "Regs [8] ( "&
    "(A[1] IS (7) RESETOUT(lrst) CHRESET PULSE0 ) "&
    "(B[1] IS (6) RESETINPUT(lrst)  ) "&
    "(C[1] IS (5) RESETINPUT(lrst)  ) "&
    "(N[2] IS (4 DOWNTO 3) NOUPD ) "&
    "(D[1] IS (2) RESETINPUT(lrst) NOUPD ) "&
    "(Y[2] IS (1 DOWNTO 0) NOUPD ) "&
    "      )";
```

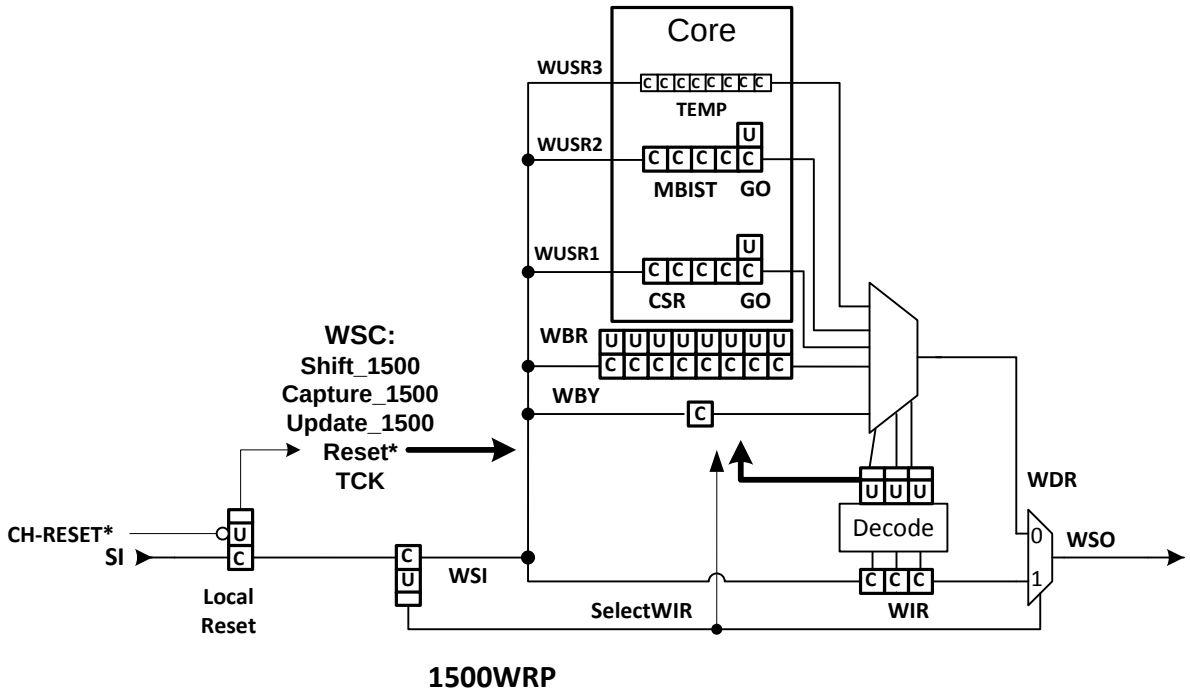


Figure 1. 1500 Wrapper

Figure 1 illustrates a small 1500 architecture. There could be hundreds of these within the IC. Without local reset, then CHRESET* or RESET* would be distributed to all of them and all would need to reset in unison, which is only desired at power up. One can see from this that 'scanning in' the reset value is

not practical, as each WUSR, WBR has to be scanned. When local reset is provided by 1149.1, 1500 structures get a new advantage. The single CORE can be reset separate from all the others, so those cores going through MBIST for instance are not interrupted. A new 'safety' is provided with local reset such that after a series of PDL based tests, the entire wrapper, including the WIR can be reset without disturbing all the other 1500 wrappers undergoing test.

Local Reset _control

a) Shall include one of the reset types such that the TDR bit is reset at power up.
b) Shall be of TDR cell of type PULSE1 or PULSE0.
c) Shall include a RESETOUT type with a unique RESET name.

d) Recommend RESETOUTPUT TDR bits use CHRESET, PORRESET or DOMRESET

- Note: RESET independence is lost if a local reset_control bit uses TMSRESET as all RESETINPUT TDR cells with the same RESET name will also reset.