Relieving the design headache

Ten steps to ease the process of implementing multivoltage chip designs. By **Arvind Narayanan**.

Multivoltage designs are increasingly common in ics for mobile devices, but can be difficult to implement. The design flows for multivoltage architectures are inherently complex and present a plethora of new challenges because many blocks are either operating at different voltages or are shut down intermittently.

Specials cells such as level shifters and isolation cells need to be used on nets that cross domain boundaries if the supply voltages are different or if one of the blocks is being shut down. Meanwhile, multithreshold cmos switches involve switching off certain portions of an ic when that functionality is not required, then restoring power when that functionality is needed.

Physical design tools and best practice flows are beginning to catch up with the demands of multivoltage design and use of the latest tools can make the difference between taping out the design on time and within specification or facing the consequences of a late or underperforming product.

As we all know, any complex problem can be broken down into smaller problems. The 10 steps presented below represent a simplified flow for physical design of low power ics with multivoltage domains. In reality, before embarking on a multivoltage design adventure, the system architects must be positive that multivoltage is indeed the best option. Can the power budget be met through the use of multiple threshold cells or clock gating? Is it better to use hardware/software partitioning? If so, then stick with those techniques. If not, arm yourself with the latest information, and proceed with caution along these 10 steps.

Architecture selection

Ensure that the architecture is frozen and captures all the power constraints required for the chosen multivoltage style in the IEEE1801 Unified Power Format (UPF) file.

RTL verification and synthesis

Using the UPF file, perform thorough power aware functional verification. This will uncover any number of functional bugs, including failure to retain state information, improper sequencing, reset failure, activity in OFF states and, occasionally, an unexpected 'always OFF' condition. Ensure that the simulation and verification runs are complete and validated, since errors here will lead to painful engineering change order iterations. Complete RTL synthesis and derive the gate level netlist.

Data import

The next step in the design flow is the netlist to gds implementation. Begin by importing all the standard physical design files – including LEF, .lib, SDC, Verilog and DEF – for the given design. Properties that are relevant to the multivoltage design flow include:

* special cells in the library, including 'always_on', 'is_isolation_cell', 'is_isolation_enable' and 'is_level_shifter' attributes

* process corners and design modes for the different power domains. Ensure that the worst case timing and power corners are setup correctly to optimise concurrently for power and timing

Power domain setup

Read the power domain definition by sourcing or loading the golden UPF file (the same one that was used for RTL synthesis). After reading the UPF file, the following items will be defined:

- domains with default power and ground nets
- power state table (PST) to define all possible power state combinations (see table 1). The PST allows designers to capture the valid combination of voltages that the design needs to be analysed for timing or power budget.
- level shifter and isolation rules for the different voltage domains
- power or ground switches for domains that are shut down

Floorplanning and placement

The physical implementation tool should instantiate the voltage islands automatically as regions with physical boundaries and create the power mesh for each supply net defined in the UPF. It should also insert special cells automatically for voltage islands and power shut off regions, including isolation cells, level shifters, multithreshold cmos switches, always on buffers and state retention cells. The designer needs to define domain specific hierarchy mapping and library association based on the architecture. The placer should group cells into the partitions and assign partition pins.

Power domain verification

This is a critical step to ensure that the power constraints are defined properly and to avoid last minute surprises. Perform checks for general design and UPF

Fig 1: Using a place and route tool to visually verify the power domains and special cell placements

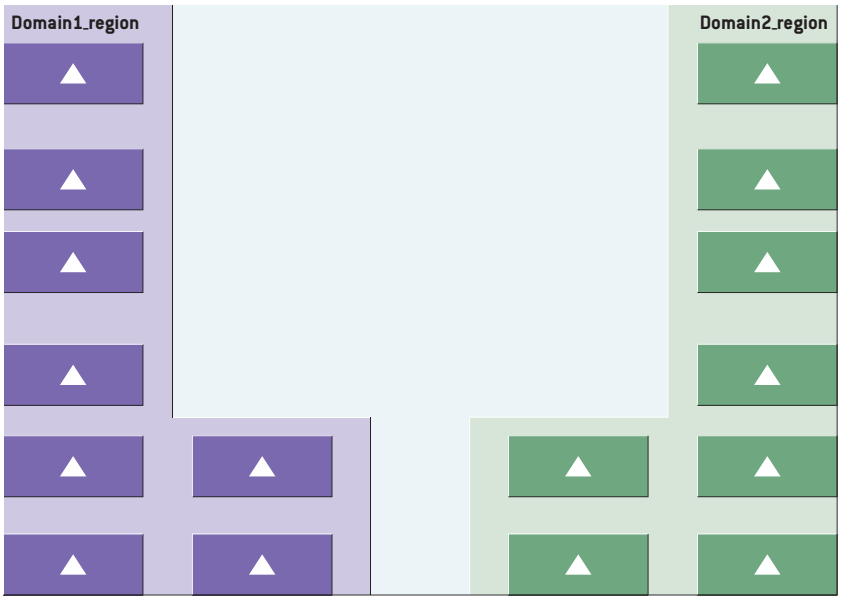

Table 1: An example power state table

Power State Table MAIN_PST (min, nom, max)

	Vdd	DOMAIN1_VDD	DOMAIN1_VDD	active
All_on_nom_v	(1.08, 1.2, 1.32)	(1.08, 1.2, 1.32)	(1.08, 1.2, 1.32)	yes
Domain1_low_v	(1.08, 1.2, 1.32)	(0.864, 0.96, 1.056)	(1.08, 1.2, 1.32)	yes
Islands_off	(1.08, 1.2, 1.32)	(off)	(off)	yes

setup; verify that level shifters are inserted on all the nets crossing different voltage domains and that isolation cells are inserted on floating nets for domains that are shut down. Analyse always on connections to ensure that the signal is buffered correctly.

The boundaries of power domains and the cells assigned to each domain should be easy to verify through the design tool graphical interface (see fig 1).

7 Pre clock tree synthesis optimisation

During the pre clock tree synthesis flow, ensure that no port punching (duplicate ports) occurs on power domain interfaces. The optimisation engine in the design tool should use the power state table when buffering nets to choose between always on buffers and regular buffers automatically.

8 Clock tree synthesis

Each domain should have only one clock port and its own clock tree network. Dynamic power optimisation in the clock tree will result in significant power savings.

Like the optimiser, the clock tree synthesis engine should use the PST based buffering solution while expanding the clock tree network. Some clock tree synthesis flows require special clock gate classes to be recognised in order to restrict sizing operations during the process to only equivalent class types. Whatever the specific requirements of your flow, the results will improve by using a multicorner, multimode (MCMM) aware flow. MCMM considers power domains, modes and corners simultaneously to reduce functional skew and to balance skew across corners. The end result is that the design should meet timing and power requirements for all the mode/corner scenarios.

9 Routing

The router should honour the domain boundaries and contain the routes within them. Secondary power pin connections for special cells should follow automatically special properties set on the power pins.

Many design flows also require double vias and non default width wires for routing of the secondary power connections. Top level nets that span across domains can be handled using 'gas stations' to help optimise timing and area.

10 Verification

After routing and post route optimisation, timing and power analysis will let you know whether the efforts invested in steps 1 to 9 have paid off. If they have not, you may need to go back to the architectural planning stage. If they have – congratulations! Proceed to design rule check and design for manufacture analysis and signoff, then tapeout to GDSII.

Using multiple voltage domains is the most effective method of achieving the lowest possible power use, but this strategy adds several new requirements to physical design flows.

Success with multivoltage requires careful planning at the architectural level and a robust methodology and toolset for all stages of the implementation. Once established, however, the multivoltage approach enables the development of ultra low power ics and SoCs for the rapidly growing wireless segment.

Author profile: Arvind Narayanan is a product marketing manager with Mentor Graphics who is active in the UPF Initiative.