Synthesis

Thursday, October 06, 2016

12:17 PM

* Timing is spec'ed out across some PVT corners - Process, Voltage, Temperature corners.
  + Ex. SSG\_0C\_0.6V
* The "Process" corner is about binning Si parts as slow parts, medium fast, very fast etc. Usually the binning is based on how many standard deviations is the bins boundary away from the mean.
* Some facts: Frequency (max frequency at which a logic can be run) is directly proportional to Voltage, and Process speed, but its relationship with temperature isn't monotonic. Above 0C, F is inversely proportional to T, but below 0C, some inversion effect happens and it isn't a monotonical relationship
* A "path" is path in logic starting with some flip flop, ending with another, passing through many flops and combo logic.
* A path is called "critical" at a given PVT, if that is the one having the most delay from start to end.
  + A clock domain's frequency will essentially be limited to 1/critical\_path\_delay(seconds)
* Standard cells and memory cells scale in delay at different rates when any one (or a combination of) P, V, T changes. So a path, that contains mostly standard cells may be the critical at a given PVT corner, but a different PVT corner, a path involving more memory cells may be the critical path.
* While going from the slowest corner to a faster corner, say, from ssg to tt (while VT remained same), or, say, from 0.65V to 0.72V (while PT remained the same), it is possible that, if a standard cell path was critical at the slower corner, a memory cell path will become critical at the faster corner. However, the other way around cannot happen - so if a memory cell path is critical at a slower corner, it will remain critical at a faster corner

Reg Vt Vf

* "Vt Vf" refers to frequency (f) and cell type (HVT, P04SVT) specification at different voltages
  + Usually they are specified only against different voltages and not at different Process or Temperature corners. We simply choose the slowest process (ssg) and temperature (which could be 0C or -40C or 105C for that voltage point) and assume that, if we can pass timing at this P,T, we will pass timing at faster P,Ts.
* Max Q point = best quiscient
* MaxQ is being found out for Xavier. For embedded it happens that Max Q = Vmin. We don't know for automotive.
* Embedded = 0.625V Vmin Vsense. Automative 0.68V Vmin Vsense
  + Timing is spec'ed out across some PVT corners - Process, Voltage, Temperature corners.
    - Ex. SSG\_0C\_0.6V
  + The "Process" corner is about binning Si parts as slow parts, medium fast, very fast etc. Usually the binning is based on how many standard deviations is the bins boundary away from the mean.
  + Some facts: Frequency (max frequency at which a logic can be run) is directly proportional to Voltage, and Process speed, but its relationship with temperature isn't monotonic. Above 0C, F is inversely proportional to T, but below 0C, some inversion effect happens and it isn't a monotonical relationship
  + A "path" is path in logic starting with some flip flop, ending with another, passing through many flops and combo logic.
  + A path is called "critical" at a given PVT, if that is the one having the most delay from start to end.
    - A clock domain's frequency will essentially be limited to 1/critical\_path\_delay(seconds)
  + Standard cells and memory cells scale in delay at different rates when any one (or a combination of) P, V, T changes. So a path, that contains mostly standard cells may be the critical at a given PVT corner, but a different PVT corner, a path involving more memory cells may be the critical path.
  + While going from the slowest corner to a faster corner, say, from ssg to tt (while VT remained same), or, say, from 0.65V to 0.72V (while PT remained the same), it is possible that, if a standard cell path was critical at the slower corner, a memory cell path will become critical at the faster corner. However, the other way around cannot happen - so if a memory cell path is critical at a slower corner, it will remain critical at a faster corner

APE: [1790881](http://nvbugs/1790881)

SCE: <https://nvbugswb.nvidia.com/NvBugs5/HWBug.aspx?bugid=200216038&cmtNo=>

Process:

1. Arch Vt, Vf at a few voltage corners. For Xavier, it is ssg\_0C\_0.6V, ssg\_m40C\_0.65V, ssg\_0C\_0.72C. This is done based on use case analysis.
   1. Note that the voltage values given here are voltage at transistor, not the IC's pin voltages (called Vsense), which will be higher.
2. Design team synthesizes at each of these corners. The synthesizing tool does some optimizations based on the Vt, Vf spec given and hence the resulting netlist at one corner will be different from the netlist resulting from a different corner
   1. Physical Design team advises the Design team as to how much of negative slack is fine - i.e. how much of ps of delay, is OK, when timing is not met. This is because, normally the Design Compiler adds a lot of margin since it thinks that during PD, delays may increase. But PD team knows how much the delay may increase. If they think that DC overcompensates (which it usually does), they will advice the Design Team that, say, if we meet timing with 20ps of slack, we are OK.
   2. If we have more slack, we have to go back and forth between arch, design, PD to either reduce the frequency requirement, or get PD to accept more slack (they will have to put in more effort during PD) or design with faster cells (like SVT cell instead of HVT) etc.
   3. Eventually when we find timing passing at chosen corners, we will choose the most important corner among those chosen corners, and give the netlist corresponding to it to the PD (remember that different corners yield different netlists during synth).
   4. PD will do placement and routing with that netlist and then run timing analysis on way more corners than the ones the Design Team tried. If timing is passing at this superset of corners, then we are good to tapeout.
3. AXI Bus
4. Monday, November 02, 2015
5. 5:43 PM

* If R5 allows only 2 outstanding trasactions, then after issuing two requests, until it gets at least one response, it will not generate another request
* Outstanding transactions buffer will be in I-NIU of that client.
* Response from the MC is considered to be out-of-order. This is because MC, when it already sees that it has a DRAM page open it first services requests for the same page and then service requests for other pages. Switching pages is latency intensive and it isn't advisable to go back and forth/
* When requests have the same AXI ID, it means the master wants their responses to be in order. If the AXI IDs are different, it means the master doesn't care
* Reordering of the responses (for requests issues with same AXI ID), can be done at some NIU. Not sure which one it should be.
  + It could obviously be done at the I-NIU the master is connected to
  + It could be done in the I-NIU in the DBB (assuming the master in question is connected to an I-NIU in a cluster NOC)
  + I guess it can't be done at the T-INU. It is possible that the master may issue two AXI transactions with the same ID, but to different clients. In this case, two T-INUs are involved. So….
* It doesn't make sense to have the Re-order buffer to be bigger than the size of the outstanding requests buffer.

* When master wants data sent that is longer than the bus data width, it can send several single transfers each containing an address and associated data - the slave will simply look at the address for every transfer and take the received data to that address
* However, it is also possible for the master to send the data as a burst transfer - only the first transfer (called first "beat") will contain the address. The following beats will only have data. The first beat awrequest itself needs to tell the slave whether it needs to INCRement addresses as it receives consecutive data or keep writing the consecutive data to the same (FIXED) address (FIFO implementations) or increment until wrap boundary is reached and then wrap around (WRAP mode)
* Burst transactions can also be sent even if the data master wants to send to slave is *not* longer than bus data width. For ex., a 64 bit data can be sent as two 32-bit data (burst transaction) even if the bus data width is 64-bits. Usually used for some reg. dump type operations.
* ARSIZE in a data transfer, single or burst, represents the length of the data (16 bits, 32-bits etc.) in that transfer ("beat" in case of a burst transaction). ARLEN, if more than 1 transfer, implies that we are looking at the beginning of a burst transfer. ARBURST tells the slave whether the addressing mode is INCR or WRAP or FIXED.
* ARSIZE is described in ARM manual as "Burst Size" - This is misleading: ARSIZE represents the data-width inside a transfer, no matter whether it is a single transfer or just one beat of a burst transfer. ARM did this because they like to confuse users as much as they can.

An ARM employee reply

Kaustya,

One way of thinking about it is as follows;

Ignoring shareability, there are three fundamental memory types, Normal, Device and Strongly-Ordered.

Load/stores to Normal memory may result in any size and number of transactions being presented to the memory system, e.g. a byte load might appear as a word read to the memory, or two neighbouring half-word stores might be merged into a single word write. In addition, near-arbitrary caching and prefetching could be performed.

Explicit load/stores to Device and Strongly-Ordered regions always produce the exact size and number of transactions, e.g. a single byte load produces a single byte read, two neighbouring half-word stores produce two distinct half-word writes.

With respect to ordering, accesses to normal memory may be reordered (within certain bounds), e.g. a load from A followed by a load from B in the program may actually be presented as read B, then read A. Accesses to Device and Strongly-Ordered always appear in the order they are listed in the program; so what's the difference between Device and Strongly-Ordered....

Device accesses are only ordered with respect to other Device accesses, whilst Strongly-Ordered are ordered with respect to \*all\* other explict load/stores. e.g  load-norm-A, load-dev-B, load-dev-C, load-norm-D could be performed as ADBC, or even DBCA; whilst load-norm-A, load-so-B, load-so-C, load-norm-D must be performed as ABCD.

Pasted from <<https://community.arm.com/thread/3696>>

Electronics General

Thursday, January 22, 2015

1:35 PM

* Simple FM Transmitter: <http://electronics.stackexchange.com/questions/104553/simple-fm-transmitter>
* Varactor basic circuit: <http://mysite.du.edu/~etuttle/electron/elect40.htm>
* Some FM designs: <http://www.philadelphia.edu.jo/academics/abusbeih/uploads/Analog_Communications_Lab/Exp_7.pdf>

Cache, I/O Coherency

Friday, July 03, 2015

12:14 PM

* When there are many processors in a system (think of a CPU cluster with 4 CPUs), each CPU will have its own cache of independent sets of cache lines. But if one CPU updates a cache line and then a second CPU accesses it, the coherency mechanism (hardware module) detecs that the second CPU is accessing stale memory and does a cache flush of that cache line (probably more than just one cache line?) from the first CPU and also invalidates that cache line in the second CPU (so that it's cache DMA will automatically go fetch the freshly updated memory content)
* I/O coherency reference to something similar when a DMA is involved. If someone sets up a DMA to transfer data to an I/O from a mem location, if the DMA is I/O coherent with the CPU, it means the coherency mechanism will come to know that DMA is going to fetch from a memory location - and if that location is stale, it carries out a cache flush of the corresponding cache lines (which were recently updated) of the appropriate CPU. It works the other way around as well (in case DMA is sending samples from I/O to mem).

Taxation

Thursday, July 23, 2015

12:15 PM

* Tax is classified into direct and indirect tax
* In India, direct tax comprises of income tax and wealth tax.
* Direct tax is tax that is given to the govt. directly. No middleman involved. Indirect tax is like the tax a shop keeper collects on every purchase and essentially accumulates the tax given by all his/her customers and then gives to the govt. So things like VAT, excise duty, service tax, customs duty etc. come under the indirect tax
* Excise duty is tax paid on manufacturing activity, by the manufacturer to the govt. But manufacturer includes it in the price of the product. So you are anyway indirectly paying the excise duty to the govt.
* Income tax is about profit, not revenue. So if I sell a house at 20L, my income isn’t 20L. If I had originally purchased the house for 15L, then 20L-15L = 5L is my income. Tax will be applied on 5L, not 20L.
* Concept of time is important in taxation. For example, income is assessed for one year and tax is applied on it. It is not assessed on a monthly basis.
* Financial year in India is April 1 - March 31
* Income of one financial year (called “Previous year”) is taxable in the next year (called “Assessment year”)
* In India, sources of income are defined in 5 categories. Salary, House property, Business, capital gains and “other”
  + Understanding nuances is important. For ex. rent from a house comes under house property, but profit from selling a house comes under capital gains.
* This isn’t govt. defined, but one can classify business using unofficial terms of manufacturing, service and trading. Trading is simply making profit out of buying-selling. So it isn’t just about share trading or bullion trading. Even grocery shop owners are considered traders.
* Capital gains is essentially profit on capital assets, i.e., land, house, car, jewellery, shares etc. In India, capital gain is divided into short and long term, the temporal border being 3 years.
* The most prominent example of the “other” category is interest income.
* Gift is considered income (taxable) if it is more than rs. 50,000. If the gift is below 50,000, it is fully exempt, but if it is even 51,000, the entire amount (not just 1000) is taxable. Gifts during marriage and from relatives is exempt.
* Agricultural income is fully exempt from tax.

Citiunlock

Friday, October 23, 2015

3:52 PM

3201149559

Data.gov.in

Wednesday, December 02, 2015

2:19 PM

API Key

5473af4a21f19e35f31e60d347b8221e

Udhaya Keerthika

Thursday, July 21, 2016

2:28 PM

Name on acc: T.Dhamotharan (Father of Keerthika. No space between T and . and Dha…)

Acc No.: 30076087695

IFSC Code: SBIN0002277 (This is State Bank of India, Theni branch)

SWIFT: SBININBB465

Mobile No. on the acc: 7305535151

Dhamotharan T 8,Moorthy street, ALLINAGARAM. Theni, INDIA 625531

Name on acc: Saravana Dinesh T. Chandrasekar

Acc No.: 5201149551

IFSC: CITI0000006

SWIFT: **CITIINBXHYD**

Mobile No. on the acc: 7757044884

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Donor Name** | **Amount (INR)** | **Contact Ph. No.** | **Contact Email ID** | **Language Known** | **Location** |
| Me (Dinesh) | 10000 | (+91)7757044884 | dinesh.thogulua@gmail.com | Tamil, English | India |
| Vivek |  |  |  |  |  |
| Raj (My Brother) | 10000 | (001) 5109936171 | rajvignesh@gmail.com | Tamil, English | U.S.A |
| Arun | 20000 | (001)7325001830 | arun.K.Rajasekaran@gmail.com | Tamil, English | U.S.A |
| Sreedhar Nandam | 5000 | (+91)9490746640 | go4ash@gmail.com | Telugu, English | India |
| Sankar Salvady | 10000 | Not Known | Not Known (Facebook contact only) | Tamil, English | U.S.A |
| Edwin Ezhilarasu | 18000 | Not Known | Not Known (Facebook contact only) | Tamil, English | Dubai |
| Siva Thoppe | 25000 | (001)248722069 | tksiva@yahoo.com | Tamil, English | U.S.A |
| Chakradhar Gajarampalli | 10000 | (001)408-239-7913 | gajarampalli@gmail.com | Telugu, English | U.S.A |
| Omprakash G. Mohanram | 10000 | (+91)7760819819 | omprakash.guniya@gmail.com | Tamil, English | India |
|  |  |  |  |  |  |
| **Total** | **118000** |  |  |  |  |