Phases

segunda-feira, 6 de setembro de 2021 14:46

02 - Digital 01 (Web view)

- **Macro phases: functional, electrical and geometrical**
  - Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhora)
  - como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Diferentemente, do ponto de vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância).
- **Geometrical phase more mature then others**
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.

Geometrical

segunda-feira, 6 de setembro de 2021 14:48

02 - Digital 01 (Web view)

- **Related to backend, layout**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar.
- **Main and most mature interaction point between teams currently**
  - não mexo muito com o backend, mas participei de coisas interessantes, como floorplan e posicionamento de pinos (LEF/DEF), qual geometria usar.
  - avançamos significativamente nesse ponto, onde temos a definição dessa geometria desde o início do projeto e refinado ao longo do projeto.
  - Também entra as ferramentas de versionamento para deixar mais transparente,
  - Essa troca aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila.
  - "Temos um caminho claro e a aderencia tá boa"
  - Eu vejo a troca entre times acontecendo no floorplan/pinos e depois na entrega, quando você integra a biblioteca OA lib, passar DRC/LVS e envia pro analog integrar
  - com todos falando a mesma lingua de DRC e LVS,
- **Chipus differential advantage**
  - Está melhorando e ainda pode melhorar, onde traz um grande diferencial para a empresa pois trabalhos com muitos serviços independente da tecnologia.
- **Bigfiles problem**
  - o problema dos bigfiles... é um ponto aberto que ninguem pegou essa bola ainda. É uma questão global que vai acontecer e que está sendo negligenciado porque não deu problema ainda.

04 - Digital 03 (Web view)

- **Related to backend, layout**
  - Deliverables trocados entre times além de informação e documentos de sign-off, eu diria arquivos de estrutura (LEF/DEF) lá no inicio do design com tamanho e pinos, apesar de que eles vão ser iterados várias vezes.
  - Depois o digital acaba trocando os mesmos arquivos: GDF (layout completo, timing fechado, power no budget, dentro da area), LEF, DEF, CDF, extração, LIB, SDF, netlist (CDL), OA lib.
- **Intesive interaction expected**
  - As interações com o outro time existe um caminho de interação que é inevitável que é o alinhamento de tracks: analog não tem visão das grids do digital, então a posição dos pinos precisa ser ajustada para alinhas nas tracks (para evitar problemas de route) e retornar essa informação pro analógico

[[OBSERVATIONS]]

segunda-feira, 6 de setembro de 2021 15:49

The development is divided in 3 macro **phases** according to the abstraction level of the circuit: **functional** (behaviour), **electrical** (transition times and loads) and **geometry** (backend and layout). Those phases are perceived with different **discipline visions** according to the actors: digital vision, analog vision, mixed-signal / top-level / systemic vision and extra-chip (like workbench or in-field operation).

The most mature interactions are related to **geometrical** artifacts, which have a more structured procedure, a better system vision and, consequently, less unnecessary overhead even though there's still a constant back and forth (mainly due port alignment with digital track grids and unclear/changing specs). **Electrical phase** have a low priority on the interactions, were digital constraints are guessed and requests for more precise data aren't clear for analog designers.

Even though all essential **requirements** are ok at the end, few non-essential ones usually are left out or workarounds are required to operate the IC as intended. The main problem seems to be related with the **interface** between digital and analog. The source of them seems to be the lack of a shared (systemic) **discipline vision**, recurring unregistered or informally-stated changes occur and divergent port nomenclature standards at the **interface**.

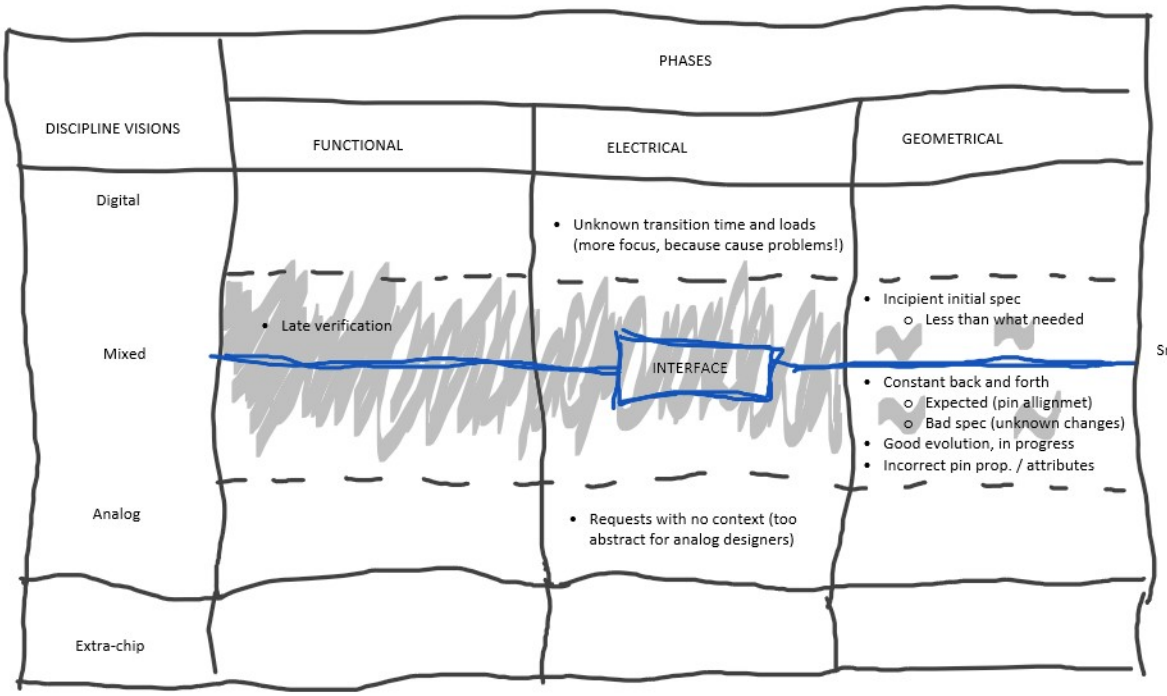
During most of the project, the other team block is considered a black box with very low specification or consideration for the other teams flow and context. It seems that both teams "stop looking" a little before the **interface**, creating a grey-area gap. More transparency and integration of the designers would mitigate and share knowledge about the project, improving the overall quality.

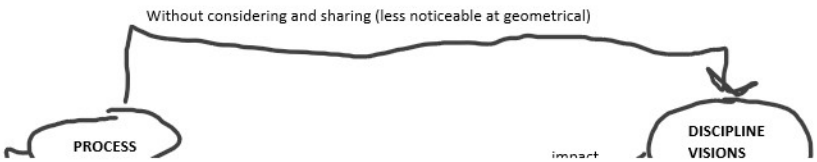
It's suggested that the lack of **formalization** is the main reason for this incompatibility between the technical flows. The usual informal exchange of information focus on specific details without considering the systemic vision, missing important context for optimal procedures. **Process** of development would have less unnecessary work and more complete deliverables considering context, but must also include time to register design/architecture decisions and update documentation for future project versions.

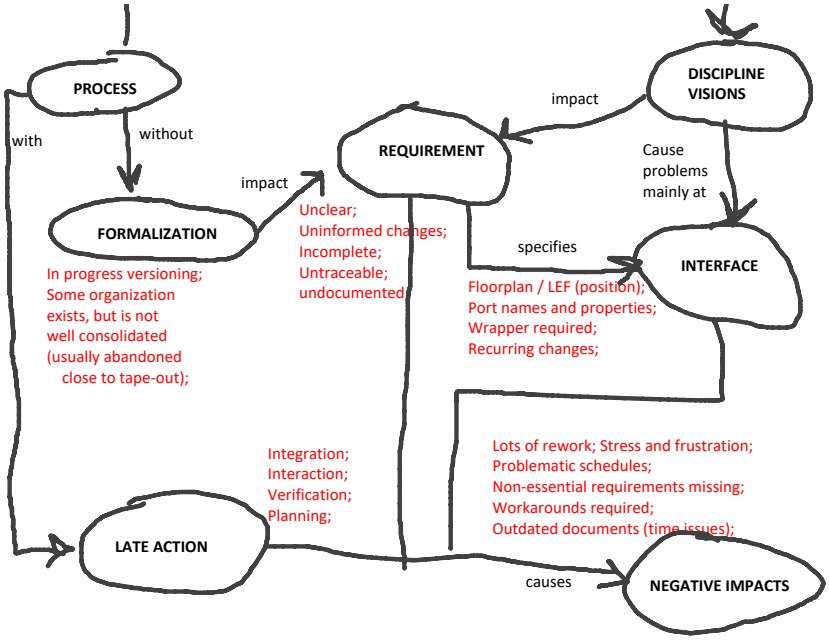
Even though some structure exists (mainly on **geometrical phase**), it's not well consolidated and dropped when tapeout date gets near. **Formalization** would create a stronger structure and bring practices to guarantee the transparency needed for a better contextualization of the process, as well as increase the focus for a more complete specification earlier in the project. The standard process documents of Chipus should be improved to reflect established procedures and create missing **formalizations** to guarantee those key points (verification, planning, specification, integration, project flow) early in the project, focusing on prevention of problems instead of inspection.

A better understanding of steps and tools of the other team will make inter-discipline deliveries more consistency and reduce the necessity to correct them. This would bring transparency and a more "DevOps" feeling, breaking the silo-like feeling that some designers seem to have.

The **negative impacts** in the project are usually schedule and non-essencial requirements. Show-stoppers that make the product invalid don't occur, but quality-of-life features often are missing or workaround are required for the correct usage of the chip. Some designer also suffer **negative impacts** regarding mental health, not only feeling frustrated for not being able to use learned skills or having to trade off quality because of time, but also constant stress caused by the tight schedules and instability of what must be done.






PROCESS
with
without
FORMALIZATION
In progress versioning;
Some organization
exists, but is not
well consolidated
(usually abandoned
close to tape-out);
impact
REQUIREMENT
Unclear;
Uninformed changes;
Incomplete;
Untraceable;
undocumented;
impact
DISCIPLINE
VISIONS
Cause
problems
mainly at
specifies
INTERFACE
Floorplan / LEF (position);
Port names and properties;
Wrapper required;
Recurring changes;
Integration;
Interaction;
Verification;
Planning;
LATE ACTION
Lots of rework; Stress and frustration;
Problematic schedules;
Non-essential requirements missing;
Workarounds required;
Outdated documents (time issues);
causes
NEGATIVE IMPACTS