

01 - Manager 01

Interview – 26/08/21

Já trabalhei com mixed-signal desde empresas anteriores, mas na atual todos os projetos são. A troca de entregas sempre trás **abordagens diferentes** por causa dessa **visões relativas de cada disciplina (analog e digital)**.

Nossos projetos geralmente são **analog-on-top** e blocos de outra disciplina geralmente são considerados como **caixas pretas durante muito tempo**, por exemplo um bloco digital geralmente fica como black-box, ou seja, **ignorado por muito tempo** em um bloco superior analógico. Isso acontece porque não existe um **modelo**, acaba esperando um RTL ou a netlist postlayout para ser enviado e considerado pela disciplina analog. Essa é uma das trocas que acontece mais ao **final do projeto**, pois temos deixado para se preocupar com modelo (no sentido de funcionalidade) bem lá pro **final do projeto**.

Outro arquivo que compartilhamos é o floorplan (abstract view em formato LEF). Nessa parte, eu acho que é preciso saber **o que vai ser feito do outro lado** e **formalizar** as definições. "OK, o digital vai ser esse quadrado", mas são **definições muito incipientes**: quais as posições dos pinos? Quais os nomes dos pinos?

Nomes diferentes são usados de cada lado (**mismatch de nome de pino**) e em alguns projetos foram necessários wrappers para **adaptar os pinos** mesmo estando tudo **dentro da mesma empresa**. Também acontece pinos com **direção diferentes**, pino de power sem propriedade de power, entre outros. Acho que acontece porque falam **"Eu sei que tem que fazer isso"**, **"Eu também sei"**, **"Eu também sei"**, mas eu acho que **falta formalizar** isso. **"Eu to te passando, se der erro me avisa"**, mas acabamos nos complicando nisso.

Eu tenho noção no que as entregas são usadas pela outra disciplina, mas **não acho que isso está claro** para todos os envolvidos. Acho que ter uma **visão do fluxo específica do projeto** corrente bem cristalizado ajudaria na clareza, pois acho que não fica claro a **utilidade de cada arquivo** para todos.

Quando chega na **verificação de topo**, informações de como ou quais testes serão feitos fica **centralizado no responsável** pela integração ou topo com **pouco envolvimento de outros**.

Que recebemos do analog, geralmente é **Liberty e LEF (abstract)**, talvez alguma coisa para simulação digital como um **modelo Verilog** do analógico, mas geralmente nós geramos ou temos algum envolvimento (nunca geraram um modelo sem auxílio nosso). Modelos nem sempre estão **bem definidos**, pois não basta saber **o que o sistema faz**, também é importante saber **até onde vamos modelar**, o que nem sempre está claro pros projetistas. É importante fazer esse **julgamento dessas questões**. Acho que **falta (nós, os gerentes) conversarmos mais** para os projetistas entenderem o que precisa nesse modelo. As vezes **não demos subsídios suficientes** para o projetistas saber o que tem que modelar.

Para refinar o que tem de ser feito, geralmente tem **muitas interações**. Como geralmente ocorrem **muito tarde**, acaba tendo que **sacrificar alguma coisa** (qualidade), como checagens ou não cobrir algo. No mixed-signal, mostra que não basta ter uma visão clara do sistema do ponto de vista do analógico ou do digital, da **visão de uma disciplina**, mas uma **visão mixed**. Um exemplo foi um projeto onde tivemos uns problemas que ficaram meio que na **zona cinza**, como uma feature que é controlada pelo digital, mas o entendimento da feature é mixed pois estava relacionado com medição de corrente. Por isso, na hora de **especificar e validar as features**, tem que ter uma **visão mixed no topo** e as duas **disciplinas tem que se entender** (as vezes até **considerar extra-chip**: dependendo dos instrumentos de medição que serão usados ou testes em silício). Essa feature foi pra **spec pouco detalhada** e, apesar de muito simples, tinha de ser visto na **perspectiva do sistema**. Features simples que você tá **cuidando desde o início** (na integração, na verificação) é mais difícil de deixar um **buraco**.

O **problema de comunicação** geralmente mostra que **não deixou claro o que tu quer receber ou o que tem de estar verificado** (não tá escrito, não tem ticket, ticket muito sumariado, definição não clara). Em termos de disponibilidade está ok, de vocabulário não parece ter muito problemas, mas tem falta de clareza do que **significa o que foi pedido**, de **como tem de ser gerado**, não sabia **pra que ia usar** e mais para esse lado... acho que essas **formalizações ajudam**. Não sei se é "Tu não me entendeu bem" ou "Eu não expliquei direito", mas tem esse lado que seria bom ter **referências para as pessoas se basearem**.

Quando faço um pedido para o analog, eu acho que as coisas se resolvem num tempo bom e de boa forma. Acho que é mais a questão de **refazer por falta de planejamento**: "Como vai ser a alimentação desse bloco?" não foi conversado antes e vai ter que refazer, mas **não porque alguém não fez direito**. Nem tudo da para planejar inicialmente, mas é importante **planejar iterativamente** para limpar mais essas coisas de refazer.

Padronização carece: ticket, onde se trocar as coisas (SVN, SOS e Redmine já melhorou bastante a localização de arquivos), mas pode vir de qualquer jeito: Google Docs, algo sem template, etc... **Falta padronizar esses pontos de hand-off**, que são os pontos chave onde se passa a bola. Padrão de informação me parece mais importante que templates ou essas coisas, pois como a **empresa está crescendo** (mais projetistas), como garantimos que todos tem **acesso a informação** de como tem que

"Eu uso disciplina para diferenciar digital e analog pois não gosto de falar que são times diferentes..."

Essa fala ressoa muito com Total Quality... É mais focado na auditoria de validade do que na entrega planejada para estar correta "de primeira"

fazer, como tem que entregar? Não necessariamente template, mas todo mundo conseguir saber **como tem de ser feito** e quais **pontos tem de ser cobertos**. Por exemplo, sempre que precisa gerar um Liberty para o analógico, ainda não está claro **o que tem que ter lá dentro ou qual a precisão**. Uma empresa para qual prestamos serviço gostaria que Liberty de um bloco analógico já levasse em conta as capacitancias extraídas. Isso é um pouco mais do que fazemos aqui na empresa, mas **precisamos decidir se queremos assim ou não**. Geralmente queremos um LIB bem simples, mas precisa ficar claro o que vamos fazer e a precisão. Se o projeto puder sacrificar precisão para ser mais rápido ou precisa investir mais por ser uma questão delicado, deveríamos **decidir isso de ser uma forma mais predeterminada** (projetista fica sem saber, fazem o que entenderam e geralmente tem de refazer algo). Ainda no LIB, se o projetista analog **não tem essa clareza ou a ferramenta** (scripts) para isso, o projetista digital **precisa gerar (ou pedir) o que ele precisa usar** ao inves de apenas receber.

Em outro projeto, precisamos fazer ECO novamente por causa de **problema com pinos** (pinout errado em arquivos). Nele, um dos pinos digitais veio como power/ground. Como era um **bloco com muito pino**, gerou um **warning (que passou despercebido)** e **suprimiu uma lógica**. Esse erro num arquivo simples é um **tipico ponto** que **não é uma preocupação pro projetista analog ("botei qualquer direção")**, mas resultava num curto quando a lógica estivesse em alto. Conseguimos perceber a tempo e, quando traçamos o problema, **era no LEF recebido** pelo time digital que um pino estava com uma **propriedade errada**.

Interface é a parte onde chega o que você não sabe ("eu conheço até aqui, tô trabalhando nessa sala"), e a mesma coisa do outro lado ("na outra sala"). Esse problemas mostram que **"logo ali tá ficando um ponto cego"**. **"Os dois pararam de enxergar um pouco antes e ficou um buraco"**.

Geralmente o impacto desses problemas é **atraso, menor produtividade**. Bug eu não sei se já foi para silício, corremos para resolver e fica **sem requisitos essenciais não atendido**. Antigamente havia mais problemas com **sinais de enable** ("achava que era ativo em alto, mas era em baixo") **ou reset** ("síncrono ou não") que eram trocados e pego apenas com **verificações de topo muito tardias** mesmo sendo **questões simples**. Acho que isso também é o ponto de **deixar muito claro no topo** porque o reset precisa ser síncrono ou etc e não definir requer adaptações de lógica, reposição e reroteamento de pino. As vezes são features não-essenciais ou que são contornáveis, então não chegou a inviabilizar o projeto.

Parece um pouco a questão de silos...
É bem "Throw it over the wall"

02 - Digital 01

Interview – 06/09/21

Em experiências anteriores, projetos eram majoritariamente mixed e havia dois times separados para analog e digital que interagiam, apresentando **problemas que tendem a se repetir**. Uma mudança de **ferramenta de versionamento** foi transparente pro digital, mas apresentou **dificuldades pro analog**. De simulação, a equipe de verificação tinha 2 colegas que centralizavam questão mixed-signal, o que era negativo pois esse **ponto de encontro tinha de ser desmistificado**.

Na Chipus, eu percebo da mesma maneira no ponto de **ferramenta de versionamento, tendo resistencia na adoção desta e de uma política de uso social**. Versionamento pode ser feito de qualquer maneira, mas uma política bem estruturada adaptando formas clássicas de fazer a coisa. Essa **convergência não é tão fácil para o time analógico**, apesar de já ter avançado bastante com as pessoas acostumando. **É um trabalho gradual**. Pessoal do digital já tem costume com isso e considera que a adoção deveria ser mais rápida, mas não é bem assim para quem trabalha de forma diferente no dia a dia.

De deliverables que vem do outro time, eu não mexo muito com o backend, mas participei de coisas interessantes, como **floorplan e posicionamento de pinos (LEF/DEF)**, qual geometria usar. Acho que **avancamos 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**, permitindo dar up, revisar, mudar posicionamento, commitar e passar isso pro colega do analog / do layout. Essa troca **aconteceu várias vezes, pessoal entendeu e aceitou, foi rápida e tranquila**. Esse é um entregável bem importante que avançou muito. Se em algum momento a coisa não andar legal, tá escapando do que era esperado, pois existe uma forma simples que tá funcionando. Esse procedimento parece estar consolidado do lado analog, mas **não sei se algum novo projetista vai saber**. Acho que **cabe os gerentes garantirem isso**. **"Temos um caminho claro e a aderência tá boa"**.

Diferentemente, do ponto de **vista funcional e elétrico (meio termo entre parte funcional e geométrica, relacionando tempo de transição, capacitância)**. Por regra (99.99% dos casos), nós não temos requisitos elétricos, um handshake, os DRVs e etc... que **"deveríamos dar uma atenção boa"**. Temos uns mecanismos para obter essas informações, mas **não temos aderência por falta de conhecimento, demora de especificação, etc.**

Do ponto de vista funcional, temos o problema de como é que se chega de uma spec para um circuito digital? **"De onde vem a spec?"** Acho que é mais o ponto de **especificação, onde e como colocar essa spec**. Tem de ser **simples para ter uma clareza**. Acho que **falta uma visão (alguém usar o chapéu) no nível sistêmico sem diferenciar disciplinas, focando na especificação sem se preocupar com implementação**. Nessa parte de spec, é um ponto que tem de ser discutido como ser feito para se ter essa **visão do topo de forma mais abstrata sem se preocupar com a implementação e alinhar melhor as coisas concorrentes para refletir a expectativa do cliente e da empresa**. Specs não realistas e etc são refinadas ao longo do projeto para facilitar a interação com o cliente. Posso ter uma visão meio miope, pois eu não tenho tanto contato com o documento de specs do projeto. Talvez nós como time devemos consultar mais o documento de spec.

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. Essa troca é interessante e que tem avançado sobre essa tarefa do lado digital, com todos **falando a mesma língua** de DRC e LVS, trazendo benefícios para toda empresa. **Está melhorando e ainda pode melhorar**, onde traz um **grande diferencial para a empresa** pois trabalhos com muitos serviços independente da tecnologia.

Entregarmos um OpenAccess versionado tem sido muito positivo, porém **evidencia o problema dos bigfiles**... é um ponto aberto que **ninguém pegou essa bola ainda**. É uma **questão global** que vai acontecer e que está sendo negligenciado porque **não deu problema ainda**. Em outro projeto já foi visto os problemas que isso pode causar.

A visão funcional (simulação de topo, mixed, etc)... **"toda a parte de frontend (funcional e elétrica) são os gargalos"**. A parte elétrica falta uma convergência (um documento, um entendimento... um pouco dos dois talvez)... é como se o **digital tivesse um placeholder mas como fazer essa informação trafega de uma forma lisa** assim como a parte geométrica (onde pegar, onde commitar, etc). Essa parte na visão elétrica **não tá clara** e talvez esteja atrelado a parte funcional, porque talvez o cara consiga **ter isso como requisito do seu bloco**. O fato da transição ou carga não ser uma preocupação/requisito do bloco analógico assim como a funcionalidade... a **questão da interface não tá claro como requisito do bloco digital**, pois tem que aparecer para quem tá projetando. Não para dificultar, só para esclarecer. As vezes é uma **informação que já tem** (qual a carga default para fazer a simulação?) **e que precisa ser passado para o outro lado** de quem não tá projetando o bloco. **Não está claro qual informação o outro lado precisa**, o que facilitaria para **identificar pontos faltantes**. Os times deveriam fazer uma **conversa para ficar bem claro esse processo e conseguir ver o que tá sendo usado, em que documento isso tá registrado e assim vamos convergir**. **"O melhor não é perguntar para o projetista, é ter um acordo de onde fica registrado"**.

A simulação tá dentro da parte funcional, e essa parte envolve outras questões. Tem uma **questão cultural** e como viabilizar uma estratégia top-down. Tá relacionado a **especificação sistêmica / alto nível**. A parte da **simulação AMS** está tentando abrir essa caixa preta em ambos times, mas precisa ter um **conhecimento em comum de simuladores, metodologias, opções**... isso tudo tem que ficar mais claro, faltando **transparencia e conversa**. Faltou essa parte de **interagir mais, trocar mais figurinha**. Isso deve refletir sendo uma preocupação desde o início no projeto, **não podendo ser deixado pro final pois sempre tem causado problemas** e não fica tão bom quanto poderia.

Comunicação entre times geralmente é tranquila na parte informal, porém **não tem um mecanismo**

Três pontos de vista relacionados com as entregas: funcional (incipiente), elétrico (muito informal) e geométrico (bem avançado com espaço de melhoria)

Os dois grandes pontos de encontro entre o time são o LEF (incremental, começando no início do projeto) do analog pro digital e a entrega OA lib do digital pro analog)

Sem prevenção do problema de bigfiles

formal para garantir que todos entendam o processo. Essa informalidade acaba focando em detalhes sem se atentar ao todo (qual o valor da capacitancia?). As vezes gera uma demanda, um esforço do outro lado enquanto só era necessario saber se o valor estava abaixo de um determinado nível, por exemplo. Sempre é importante entender o contexto dos dois lados, a informalidade sempre faz isso ficar de lado e sempre causa perda de informação. A falta de formalismo reflete uma falta de contexto. "Liguei pro fulano, ele me passou a informação", mas isso deixa descontextualizado. Talvez ter reuniões pré-definidas, checkpoints/checklists de informações a serem passadas... esse formalismo que vai ajudar nessas questões. Noto que falta trocar informações entre o time, geralmente caindo em uma conversa informal quando necessário.

Formalismo para garantir que tarefas multidisciplinares envolvam todos os atores, nem que seja uma apresentação. Idealmente, durante o projeto para dar transparência e trocar conhecimento entre participantes. Se não der, após para ter pelo menos a troca de conhecimento. Vocabulário e indisponibilidade eu não sinto muita falta. Contexto realmente é o que faz falta.

Esse problemas geralmente geram retrabalho, trabalho desnecessário ou mais complexo do que necessário, bugs tardios ou não pegos... Uma falha catastrófica não passa porque temos multiplas etapas de verificação, nós tendemos a pegar eles apesar de poder acontecer. O que acontece é afetar a qualidade do produto final (mais no sentido de problemas não essenciais, como valores default ou exigir um workaround para funcionar) ou algum problema que não conseguimos entender o motivo.

03 - Digital 02

Interview – 06/09/21

Porém já tive **muito problema com interface**. Um problema muito grande em um projeto foi os **pinos estarem totalmente diferentes no digital e no analog**. Foi necessario fazer um wrapper de topo só pra renomear as portas e rerotiar alguns indices de fio (digital usava bit 3 e analogico usava bit 20, algo assim, principalmente bits de controle). **O padrão utilizado pelo analógico era totalmente diferente do usado pelo digital**, e quando **notamos isso já era em cima da hora** então só encapa isso aí e segue o jogo. Na verificação de topo apareceu umas coisas erradas e uma discussão entre gerentes dos times foi descoberto a diferença dos pinos. Alguem fez um wrapper para fazer as correções de interface e me limitei a testar o digital para ver se estava arrumado. A entrega foi feita por SVN e indica qual revision era, **não teve documentação ou requisição referente a essa mudança**.

Teve projeto que **pinos da interface mudavam da noite pro dia, funcionamento mudando em cima da hora**. **Todo pedido de mudança era boca a boca**. Apesar de ter uma especificação inicial bem redondinho, com planejamento bem arrumadinho, mas o **digital entrou só nos finalmente**. Sempre na cabeça do analógico (e até tem um pouco de verdade nisso) eles começam o trabalho bem antes porque a parte analógica é mais complicada, então **deixam o digital pra depois** que é mais simples. **Digital entrou bem depois, começou já com a corda no pescoço com uma spec bem fechada**.

Se for ver os projetos da Chipus, uns 80% das mudanças vem do analógico e digital acompanha, acho que também por causa da característica dos projetos da Chipus, com a parte do digital sendo mais controlador, banco de registradores... um circuito mais acessório pro analógico.

Apesar de ter uma spec inicial, vem **várias mudanças e a documentação fica desatualizada**. Não sei se o gerente da digital dava pitaco nas reuniões, mas quando eu entrei já tava tudo definido e ficou vindo mudanças do gerente da analog. A gente até tinha reuniões, mas não mexia com a característica do projeto ou o que a gente precisa, só focava na implementação. Tinha uma visão do porquê, uma visão de topo.

Eu conversava com o pessoal do analógico, pessoal ajudava, explicava bem direitinho, era bem tranquilo. A linguagem tava tranquila, conseguimos nos entender de boa. Como os circuitos da Chipus são mais acessórios, eram geralmente para tirar dúvidas, confirmar protocolos de comunicação, confirmar geração de estímulos de acordo com o que eles estavam planejando... não lembro de pedir uma mudança devido a dificuldades do lado digital. **"Analógico é mais a parte pensante do projeto, o digital é só um acessório, meio que já tá tudo definido por eles ... É mais do lado deles a função de arquiteto"**.

Tinha muita conversa, mas não por falta de clareza na explicação. Geralmente uma dúvida ia puxando outra... estava com dúvida num sinal e depois via que esse sinal influenciava outra coisa. Acabava abrindo uma árvore de dúvidas.

Em um projeto, eu implementei e verifiquei, o que já não é um bom começo. **A verificação começou bem tarde**, só deu tempo de fazer testes diretos. Em outro, eu já entrei no incêndio, fora do tempo... **Verificação nem foi finalizada**... Não deu nem tempo de fazer verificação postlayout, apenas RTL. Em outro começamos bem adiantado, mas quando voltei pra verificar a outra parte **já tava na correria e tinha que aumentar o ambiente** com não só novas funcionalidades, era um mundo completamtnete diferente com modos de operação. **"Pelo menos a parte de verif, nunca tive uma experiência boa com a Chipus, sempre correndo pra apagar incêndio"**.

Geralmente causa os impactos negativos de cronograma... Pensando do meu lado, **não é desconfortável... é uma palavra mais negativa que isso**. Aquele sentimento de que não tá fazendo trabalho bem feito, porque **é correria, estresse... quer colocar o que tu aprendeu na prática mas não dá na vida real, vai o que der**. **Saúde mental do trabalhador: cara vai dormir e fica pensando naquilo, acaba sendo muito desgostoso**.

Acaba ficando muito bagunçado, aí quando vai **retomar um projeto a documentação tá toda desatualizada**. Aí fica tentando lembrar de memória porque algumas decisões foram tomadas aí no meio de uma reunião você vai lembrando de repente as coisas. Esse replay para concertar bugs da versão anterior com documentação desatualizada tem o problema de pessoal que saiu da empresa e aí nem essa conversa dá para ser feita direito e fica informação perdida. **"Tem uma black box no meio do projeto"**, **"Mas o que é isso aí? Vamos tirar"**, **"Não, tira não que tá funcionando"**. Acaba atrapalhando muito essa **falta de conversa inicial, de planejamento, de cronograma e digital não estar a par da conversa**.

Parece indicar que, apesar de disponibilidade, não existia descrições suficientemente completas para compreensão do sistema

Impacto no projetista! Não so no projeto!
"Tanto de vezes que fui dormir e sonhei com código"
"É aquele estresse o tempo todo, correndo de incêndio Atrás de incêndio"

Aqui conversa com aquele documento de Architecture Decision Record (ADR) que poderia trazer benefícios nessa questão

Acaba faltando um pós-projeto para deixar a "casa arrumada".
"Não é raro ter uma rerodagem", então tem que considerar que o projeto vai voltar então tem que deixar um tempo depois do projeto só para documentar ao invés de já jogar em outro projeto

04 - Digital 03

Interview – 08/09/21

Participado de projetos na Chipus esporadicamente, entrando perto da data limite mas de forma mais controlado, não pagando incêndio. Um projeto corrente possui conversas semanais. O time do analógico geralmente se reúne com o topo e definem coisas, que são abstraídas pra gente e não temos muita interação direta com eles. "Só pegamos coisas prontas e consequências de reuniões desses caras".

Dos últimos projetos eu tenho uma boa visão do topo, com constante comunicação com o analog. Mas acho que foi sempre bem conversado e é transparente, nunca foi algo muito atômico, sempre precisando saber um pouco mais do contexto: onde ele vai ser integrado, o que tem a gente vai entregar, pra quem, o que aquele deliverable vai fazer. Tem um limite do quanto sabemos do topo, vou até onde influencia o bloco que estou fazendo, raramente vou para aplicação ou entender o contexto de onde o chip vai ser usado. É diferente em outro projeto em que estamos fazendo um IP, em que precisamos saber tudo para montar um test chip.

Receber/entregar coisas com time analog é meio desorganizado, não tem muito método... existe muita comunicação verbal (principalmente com poucas pessoas envolvidas): request, faltou um arquivo, um deliverable, tenho que avisar que tá pronto e explicar como pegar,... Acho meio informal essa comunicação. Já evoluímos muito nisso como documentos (sign-off document), mas eu diria que sempre tem uma hora perto do tape-out que vamos para o verbal demais. Acaba não se sustentando mesmo quando tá muito organizado e acaba indo pro verbal: "Faltou uma coisa", "Tem alguma coisa errada", etc. Tem que se pensar como organizar mais isso, mesmo quando no modo "apagar fogo". Evoluiu muito desde que entrei e acho que isso se deve a estrutura que tem para entender de onde as pessoas devem pegar ou colocar as coisas.

Deliverables trocados entre times além de informação e documentos de sign-off, eu diria arquivos de estrutura (LEF/DEF) lá no início do design com tamanho e pinos, apesar de que eles vão ser iterados várias vezes. Já chegamos a trocar SDC, porém é mais complexo pois o pessoal do analógico acha meio abstrato. Pro digital, seria bom entendermos o que tá do lado de fora para entender a conectividade para deixar o bloco mais real e menos a gente arrisca. Isso é uma coisa que deveria ser trocado mais, mas acho que não acontece muito. Depois o digital acaba trocando os mesmos arquivos: GDF (layout completo, timing fechado, power no budget, dentro da área), LEF, DEF, CDF, extração, LIB, SDF, netlist (CDL), OA lib.

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 alinhá-los nas tracks (para evitar problemas de route) e retornar essa informação pro analógico. Outros tipos de interações poderiam ser evitadas com maior maturidade da especificação. Tive muitas interações de deliverables porque adicionaram, removeram ou mudaram pinos.

Problemas de integração já vi por causa de versionamento: versão de floorplan que não esteja alinhado com o topo (interface modificada). Versionamento entre deliverables relacionados também tem muito problema, seja intrabloco (incompatibilidade de parasitas, em uma versão mais antiga, com o design atual) ou interbloco (entrega consistente entre todos os deliverables de subbloco). SDC também costuma ser um problema... como não damos muita atenção chutamos um valor e na integração vemos que fugiu muito, acabando que tem violação de timing, capacitância, power... é um grande problema de integração essa definição de interface.

As entregas do time analógico são rápidas, mas são muito vindas do acordo verbal. Como o

analógico não trabalha com versionamento, que acho que é uma questão de educar todo mundo junto de trabalhar com versionamento, o que é difícil para eles pois não trabalham com arquivos "version-friendly". Isso acaba causando muita troca boca-a-boca e quebra nosso versionamento. Já tivemos problemas de definição de pino, em que um pino de sinal foi definido como de power no analógico, o que só foi detectado ao final do bloco. O fato de fazermos essas trocas de forma verbal é rápida, mas acaba sendo pobre.

Quanto a comunicação com o gerente do analog, a solicitude é perfeita. Geralmente explicamos o que precisa (ultimamente não tem precisado mais), sempre foram solícitos e sempre recebia o que eu queria do jeito que eu queria. As vezes tem muita conversa para chegar num meio termo (mexer os pinos por exemplo). Nunca tive problemas na Chipus com isso aí. No meu caso, não costumo falar diretamente com projetistas analog, a não ser quando a geração de uma entrega apresentou um detalhe muito pequeno, caso contrário falo com o gerente. A hierarquia é bem obedecida.

Já ajudei um pouco num projeto antigo na verificação de topo. Adaptei testes digitais para podermos testar em nível analógico, além de fazer testes em FPGA. Mas integração e verificação de topo vi acontecendo mais bem pro final do projeto. Tínhamos mais preocupação se a gente ia conseguir construir ou não o que era requerido do que fazer a verificação do projeto em si. Inclusive a maior parte da verificação foi feita após o tapeout do chip.

De impacto negativo no projeto, o pior nunca aconteceu (perder funcionalidades, perder tapeout ou perder chip no tapeout), mas atraso de cronograma acontece demais. Um projeto atrasou muito tempo porque foi achado problemas encontrados tardiamente após a integração e verificação de topo.

Hoje em dia, eu cheguei num ponto que eu entendo que isso é comum. Em um projeto, é comum voltarmos a implementação do zero por causa de um erro de especificação, nova netlist, erro de integração no topo, etc. Antigamente achava frustrante, mas hoje meu trabalho circula muito entre construir a base, para que seja repetível até o ponto final do projeto. Não é comum voltar do zero e dar tudo errado, vai convergindo e tal... com isso em mente, eu não me frustro mais. "São só ossos do ofício mesmo... faz parte da vida do designer".

Em um projeto, eu fiz uma parte de verificação de power: fazer uma análise de power grid para ver se ela tava OK sem violar EM, IR drop. Teve muitas interações para corrigir isso, específico desse projeto porque tinha uma corrente muito alta. Teve um sofrimento por causa de misscommunication de especificações: tínhamos reuniões semanais para definir, entendíamos errado e acaba tendo que fazer outra coisa na próxima semana. Faltou algo bem definido do lado do cliente e acaba virando bem mais verbal, mas fica difícil fazer muito sendo do lado do cliente, acaba tendo muita conversa verbal mesmo.

05 - Digital 04

Interview – 09/09/21

Em um projeto fiz setup de scripts e entregas de umas macros a serem usadas dentro do digital e uma entrega de layout. Um problema de iteração com o time analog era que nós do digital usávamos SVN (versionamento) e tínhamos que deixar em um local público para o analog pegar. Também tive que pegar um arquivo deles, mas ficou confuso onde pegar e qual versão aquele arquivo era... "Pergunta para um, pergunta para outra para saber se é aquilo mesmo". Era alguma coisa envolvendo LEFs... não lembro bem. Uma coisa que me incomodou é que tive que copiar o arquivo... "Como eles vão saber que é a versão certa?", "E depois se tiver que atualizar, e se eu esquecer de atualizar lá?"... SVN é tudo muito fácil. No meu caso, não teve problema, mas duvido que já não tenha tido problema.

Quanto a visão de topo, nesses projetos eu não participei do kick-off dos projetos e pra mim era tudo uma caixa preta. Eu sabia o escopo digital e era isso. Um colega do backend reclamava muito dessa questão de integração, muitas vezes não sabia do que tava do outro lado. Ficava muito "faz assim porque é assim e pronto", era uma coisa de certa forma imposta. Mesmo tendo problemas do nosso lado, a gente recebia o input e "tem que ser assim se não causa problema do outro lado, te vira".

Não participei de nada do topo (integração, simulações, etc), fico curioso de saber como que é porque pra mim é um black-box gigante.

Não tive problemas de interface e as entregas que recebi estavam todas ok, sem precisar regerar versões ou algo assim. O meio-campo dos times era feito pelo gerente digital, geralmente sendo ele que trazia as informações e, inclusive, ele parecia mais interessado em puxar informações do que o time analógico prove-las. Geralmente o gerente digital já tinha as informações na ponta da língua ou ia buscar a informação. Inclusive "era bem comum ver ele sentado do lado do pessoal de layout para ajudar a definir, conversar e debater, fazer bem esse meio de campo entre as equipes". Ele também acabava sendo meio de campo até mesmo entre o pessoal da digital.

Eu não sei se é sempre assim, mas vi muito o pessoal do analógico definir e o pessoal do digital só segue. "Já vem com LEF, com DEF... já vem tudo pré-pronto". Pode ser só questão de fluxo de projeto, ainda mais que a Chipus começou como empresa com foco analog, talvez seja uma herança disso... não que seja um problema, é só uma característica. "Talvez muito da cultura de antes de ter um time digital, terceirizar um serviço".

Nos projetos que participei, não precisei ter interação com pessoal de analógico, só para pedir para liberar licença. Aconteceu bastante apesar do pouco tempo que participei.

Teve uma questão de um projeto que teve que regerar uns arquivos (DB->LIB, Milkyway->LEF) bem depois do projeto ter terminado por necessidade do cliente. Parando para pensar, era uma situação meio estranha... não era topo, mas ia ser passado para o cliente. Não sabia o porquê. Não cheguei a ter maiores problemas relacionados aos projetos, mas essa questão de regerar os arquivos deve ter sido por algum mismatch. Mas estou divagando.

Outro projeto participei apenas revisando DRC e LVS de outro designer digital, e tive bastante problema de compartilhamento de servidor. Não entra tanto nessa questão de integração, mas problema de infraestrutura da disputa de recurso. Entre pessoas dentro do projeto, entre projetos... a "integração de informação". Não tem um escopo bem definido de quem vai usar a máquina, quanto vai usar... Vou usar um servidor e tem outra pessoa usando, aí os dois podem ficar travados... Esse problema já aconteceu algumas vezes, seja por licença ou servidor

Da parte técnica, eu não me frustrei por falta de conhecimento ou algo assim, sempre me ajudaram e foram receptivos para me auxiliar. Mas a parte dos recursos é algo que realmente é estressante. Você quer rodar algo e não tem o recurso (máquina/licença). Pra mim é o que mais incomoda é precisar rodar algo e não ter o recurso. "É como se eu estivesse na correria e alguém falasse 'Não, para, fica paradinho, espera meia hora'". Entendo que pode existir prioridade e compreendo, mas é algo que atrapalha. Problemas em entregas não me incomodam.

Recomendou uso de nuvem e mais servidores

06 - Digital 05

Interview – 13/09/21

Entrei num projeto com backend bem complexo com várias particularidades técnicas. **Faltou bastante especificação do projeto**, tive que usar outro projeto parecido como base para entender o sistema. Foi bem desafiador e já tinha uma estrutura de scripts, mas muitos buracos. A parte de DRC e LVS tive grande apoio com o time analog, fluiu muito bem apesar do home office.

O fluxo digital foi relativamente tranquila, mas os detalhes da tecnologia foram desafiadores, Tivemos que fazer DRC e LVS apenas do pad ring para entender questões de nomenclatura. **Teve novas rodadas de trabalho** após a primeira "entrega" depois por questões de EM e corrente que foram encontradas posteriormente.

Vieram problemas que já não eram pra ter em uma segunda versão do projeto. Era para ter um fluxo bem fundamentado e acho que um grande problema foi **peessoas que trabalharam anteriormente no projeto terem saído**. Não tinha um documento com a especificação de projeto, não foi deixado um documento explicando o que foi feito e suas justificativas... É o problema de quando você entra num projeto e não sabe em que ponto está.

Foi necessário fazer ECO do IP em si. **No final** do DRC, foi encontrado um **problema no abstract em que alguns pinos estavam com propriedade errada** e ficaram desconectadas, precisando da **ajuda do gerente para encontrar**. Tivemos que fazer o LEC com auxilio de outro projetista digital.

Conversa muito com ADR

Também tiveram **vários problemas do cliente** que foram encontrados.

Faltou um documento bem explicito explicando o todo, por mais que seja simples. Tive muita **ajuda do gerente** e **estudar código para entender** o sistema. A **spec** basicamente foi via **LEFs e verilogs gerados por mim** a partir de esquemáticos do cliente. Foi o que usei para entender o topo, suas instancias e sua lógica. **Timing e power** foi bem complexo.. Já existia SDCs, mas tivemos um **grande esforço com interação do cliente**.

A memória deu muito trabalho, muito complexo, **travou máquina**...

Com o **time analog** da Chipus, eles me **ajudaram no DRC e LVS** basicamente. Pegamos muitos problemas de ambos e o **analog ajudou muito** neles. Teve muita **regrinha não sendo considerada** no DRC, várias **adaptações de nome** nas correções de LVS. Teve uma grande dificuldade, e acho que pode ter sido por causa dessa hierarquia do projeto **não ter sido muito bem feito**: usar pad antes de fazer um LVS só dessa hierarquia, LVS só da memória, problemas de nome (falta de padrão de nomenclatura).

Não participei da parte final do chip. Fechei só a parte do digital, com parte de DRC e LVS muito mais tocado pelo time analog.

A comunicação com o time analog foi bem tranquila, fizemos várias reuniões.

Não teve verificação de topo, simulação com tempo, essas coisas. Isso era responsabilidade do cliente. **Alguns problemas encontrados de ultima hora poderiam ter sido arrumados com análise de power mais cedo**.

Os impactos no projeto foi **cronograma**. O **cliente aparentou ter problemas de sincronia entre seus times**, tendo mudanças de postura na comunicação (hora com pressa de terminar o projeto, outra hora paciente). Problemas nas partes na versão anterior do projeto, teve uma certa **inocencia de que foi feito correto**, faltou uma **confirmação de**

que realmente tava certo por ferramentas. Esse intervalo no projeto anterior causou muitos problemas.

Não senti nenhum impacto negativo em mim. Pelo contrário, me senti muito a vontade, não me senti sobre pressão. Todos se ajudavam.

07 - Digital 06

Interview – 16/09/21

Nos projetos, nunca vi essas simulações de topo, apesar deles comentarem muitos de como são longas e demoram. Mas nunca me interessei que tipo de simulação tava rodando, o que tava se passando... Não tive contato com a integração também, apesar de ter tido debate de tamanho do bloco, entradas e saídas, isso teve bastante.

Em um projeto antigo, eu não tinha ideia nenhuma do topo. Eu entrei em uma nova versão de um projeto, mas não tinha noção nenhuma do porquê, do que servia, para que servia no topo... Em um projeto mais novo, tínhamos reuniões de follow-ups quase diárias mostrando o que cada um estava fazendo, inclusive compartilhando tela para mostrar e tal.

Em um projeto cuja interação era com o cliente, tinha um intermediário digital ao invés de falarmos com o time analógico diretamente.

As entregas geralmente eram recebidas com GDS, OpenAccess... passavam um floorplan e eu ia lá com um script pronto para exportar LEF/DEF. É basicamente isso que eles passam. Para devolvermos geralmente usávamos OpenAccess também.

As entregas geralmente se davam bem, mas as vezes era difícil de entender o que o outro tava pedindo, já que era de outra área e usa termos diferentes.

Mas num projeto antigo, eu tive que mexer no SOS do time analog, o que estava desalinhado e complicou as trocas. Em um projeto mais novo nós não tivemos que mexer no SOS.

No projeto mais novo, eu identificava que tinha pinos flutuando e coisas que não estavam legais. Outras pessoas do digital faziam RTL e síntese, enquanto alguém do analog gerava o abstract do floorplan. Eu cuidava do place and route e aí eu notava que tinha coisas que estavam diferentes entre abstract e RTL, como os nomes dos pinos. Trocava-se em um lugar e não trocava-se em outro, essas mudanças não eram informadas. Versões novas que não eram informadas também causavam retrabalhos.

Parece que é só rodar o fluxo de backend, mas não é tão simples, as vezes precisa de correções manuais ou mudanças no fluxo constantes.

Pessoal do analógico geralmente são muito receptivos e ajudam quando pedido. A comunicação geralmente é boa.

Em um projeto, a interação de mudanças na posição dos pinos pedidos pelo analógico era algo esperado devido as tracks do digital. Apesar de já ter sido discutido esse ponto anteriormente, parece que não ficou claro ainda. As vezes demora um pouco para ficar claro o motivo dessa alteração, não entendendo porque tinha ficado diferente do que eles querem. Também aconteceu o contrário: me pediram algo que até agora não entendo bem o porquê, então tive que perguntar para outras pessoas do digital até entender o que eu tinha que fazer.

Falta de encher o outro lado

Nas reuniões de follow-up com ambos os times, a maioria dos problemas eram bem discutidos e eram interessantes dando para entender, mas tinha muita coisa que não tinha explicação apesar do pessoal do analog se entendia. Nós da digital depois conversávamos e tínhamos essa dúvida, apesar de não afetar diretamente o nosso trabalho.

A parte de topo pareceu ser bem planejada. No projeto antigo era uma nova versão de um projeto antigo teve pouco retrabalho, mas em um projeto mais novo teve algumas mudanças mas pouco. O topo estava bem organizado e os problemas que foram reportados foram mais por causa dos desafios de encerramento, não eram modificações grandes.

As reuniões de alinhamento foram mais no final do projeto. Talvez tivesse sido legal iterações menos regulares no início para troca de opiniões.

Não senti impactos negativos no projeto ou em mim. Teve uma época no projeto antigo que estava bem chato porque eu perguntava sobre uma modificação vinda do analógico que causava retrabalho. A resposta era uma explicação que não fazia sentido pra mim, eu continuava achando que não era necessário. As vezes era só porque o cara do analógico não queria refazer

do lado dele, querendo que o digital se adaptasse no lugar. Isso ficou bem claro em um projeto mais novo, em que no ultimo bloco teve muita mudança no fluxo e não teve explicação, "digital que se adapte à mudança".

"Constraints", essa interação é de extrema importância e para definir as restrições da interface. Como o pessoal analógico geralmente sabe o tipo de circuito que vai ficar pendurado na interface com o bloco digital, parte das restrições poderiam chegar no digital sem que tenhamos que ficar "chutando" valores aproximados e, às vezes, errado. Como delay e capacitâncias nas entradas e saídas. Outro dia, um analog boy até me perguntou como era gerado o SDC, sendo que ele nem é gerado e sim baseado na tech e informações do topo...

08 - Manager 02

Interview – 20/09/21

O gerente tem responsabilidade do topo, mas ultimamente temos **treinado algumas pessoas para tomarem conta do topo**. A topo eu acho que **ninguém que pegar**... a gente tem que ter um projetista **alocado no começo do projeto** para focar no topo, não deixar o gerente... estamos tentando mudar isso. A ideia é o **gerente só verificar**, não fazer o topo.

Os projetistas dos blocos geralmente **interage com o topo só se pedido, se está afetando sua "caixinha"**. Acho que é algo que tem que mudar, mas antes disso tem que fazer os **projetistas (analógicos) olharem para seu próprio layout**, pois existe pouca interação nessas etapas de verificação: projetistas faz o esquemático e espera o layout pronto do layoutista. Porém ele **tem que participar das decisões e da verificação**, não só entregar o esquemático com notas. Tem acontecido retrabalhos... **tem que olhar pros dois lados**.

As entregas evoluíram bastante, principalmente quando o digital começou a entregar OA libs, passando algumas responsabilidades dessa entrega pro digital. **Do ponto de vista do analógico, "usamos o digital como se fosse um IP para ser integrado"**. **O signoff estar dentro do time digital solucionou bastante coisa**.

Acho que **falta mais conhecimento do digital e experiência de verificação física com ferramentas do fluxo analógico**. Quando tem que passar pro GDS e verificar DRC/LVS tem alguma lacuna de como fazer isso, pois o pessoal de layout que faz isso diariamente. **Nessa troca que geralmente contem o maior delay, não entender o que tá acontecendo, demorar para debugar**... Acabam exercitando esse fluxo só **no final do projeto**, diferente do pessoal de layout que usa desde o início. **Tem que se falar mais** entre esses dois times nessa parte de layout analógico e backend digital para resolver esses problemas.

Existem 4 entregas: **símbolo, abstract, esquemático e layout**. Analog faz símbolo e abstract para o digital, digital faz esquemático e layout (que são importados pelo time de layout). O analog faz símbolo e abstract (às vezes gerando o LEF diretamente).

O **fluxo de entregas eu considero que está aceitável**. O que pode acontecer é não saber fazer signoff direito na ferramenta, não usar os mesmos switches na verificação física... "eu entreguei, tá pronto" e quando o pessoal do topo **integra dá algum problema**...

Outra coisa que acontece é que às vezes precisa de uma terceira interação manual para passar DRC/LVS. Às vezes o **bloco digital não vem pronto** para rodar LVS. O digital pode fazer, mas precisa aprender a usar a ferramenta do fluxo analógico se não fica sem verificar até o final. Às vezes precisa colocar um guard ring por exemplo. Essa parte final **(signoff de DRC e LVS) é sempre o gargalo**, pois o **digital depende do analógico** e fica tendo **interações para corrigir** erro.

Tem acontecido (hoje menos) um estresse porque analog pede algumas mudanças pro digital, eles falam que é simples de mudar e tal, mas **tem esse passo manual** que tem que ser feito depois e o **layout começa a reclamar porque entregou o digital em cima da hora** (e tem mais trabalho por ser manual e tem que verificar)... **essa interação no final do projeto não é boa**, pois ficamos sobrecarregando o time de layout porque o **digital não consegue fazer essa verificação antes**.

A comunicação às vezes fica meio atrapalhada por um não conhecer ferramenta e fluxo um do outro. **Ultimamente atrapalha menos porque estamos entendendo a limitação de um lado e do outro**.

Agora em respeito a comunicação, eu acho que sempre temos que trabalhar um pouco mais, inclusive gerência incentivar. A pessoa pode achar chato, mas **tem que se comunicar, pedir, tentar resolver os problemas**... Temos que trabalhar projeto a projeto, pessoa a pessoa. Acho que é mais parte da **gerência desenvolver e incentivar essa capacidade do pessoal**. Acontece de ter estresse no final mas ultimamente tem melhorado... se o projeto tá dentro dos trilhos, não tem muita pressão... a comunicação acaba rolando bem.

Questões de topo são geralmente mais deixadas **para o final**. Se não tiver alguém encarregado pelo topo, vai ficar para o final porque o gerente começa tratando os problemas dos **projetistas que pergutam muito desde o início** porque já **começamos projetando**, não tendo **planejamento** ou planificação, pensando como ia fazer... pessoal consome muito tempo do líder de projeto.

O ideal seria ter **alguém responsável pelo topo** e modelar o sistema antes dos projetistas começarem a implementar. O que acontece, na maioria das vezes, é que **temos que já começar a projetar (implementar) e construir o topo em paralelo**. Não tendo um responsável pelo topo, as atividades não são adiantadas e fica tudo pro final. A ideia seria alguém responsável principalmente pelo **modelo, simulação de topo/mixed, estratégia de teste/debug e integração**. Não precisamos ter diretamente uma pessoa que ataque tudo isso, temos que juntar pessoas para desenvolver essa habilidade e capacite para ter independência (sem precisar ficar perguntando muita coisa).

Os impactos geralmente são prazo... o atraso sempre acumula e **quem acaba sofrendo mais é o layout analógico e o backend digital**. A etapa de **desenvolvimento geralmente dura mais do que o planejado**... faltou o planejamento de quanto vai demorar fazer esse desenvolvimento, simulação pode ser muito longa, não sabe como fazer, não consegue simular, simulação demorou mais do que pensava... Demora para gerar o extraído...

Pouca atenção no início de projeto aos erros de DRC e LVS...

Também existem projetos com **novidades que desconhecemos**, tendo um processo de interação de aprendizagem bem **maior do que o projeto comporta**

Já melhorou, estamos tentando fazer simulações já conectando os pads para adiantar alguns problemas...

O impacto em mim é mais a sobrecarga, acaba acumulando muito e você tem que decidir... ou deixa a

Interessante... existem "silos" dentro do próprio analógico... a divisão de tarefas em cargos parece gerar um silo sempre, mesmo dentro da mesma disciplina

Essa mudança do OA lib parece ter muita semelhança com aquela questão de DevOps do time de desenvolvimento já realizar algumas coisas de parte operacional

pessoa se resolver ou ajuda. Acaba sobrecarregando, os problemas escalam e as vezes não temos tempo para resolver. Ainda mais a parte técnica, fazendo simulação a noite, resolvendo problemas mais tarde.

Existem riscos que sabemos que vão acontecer mas não tem o que ser feito, **como mudanças de escopo do projeto**. Sempre vão especificando e incluindo mais coisas durante o projeto, tentamos minimizar mas sempre vai ter essa questão. É um problema em outras empresas também, é algo normal. São coisas que foram mapeadas, mas são **as incertezas que não sabemos ou demora para descobrirmos como resolver**. Porém precisamos **reduzir o tempo e custo de RFQ**, pois não queremos gastar recursos antes de fechar um projeto.

Acho que os principais pontos são essa **especificação inicial** e tentar ter uma **interação de digital e analógico mais cedo**... que acho que é **dificultado pela questão do topo não ficar bem definida**. Seria bom fazermos modelos digitais para termos simulações de topo. Pessoal do analógico tem aprendido mais sobre modelagem digital e suas ferramentas... **projetista tem que ser responsável do seu bloco, não pode jogar isso pro time digital**. Tem que conseguir fazer o modelo digital, mixed-signal e as entregas da sua parte.

Soluções para reduzir impacto de mudanças seriam bem benéficas. Porém existe um tradeoff de entender bem o projeto antes de começar (RFQ) e gastar pouco recurso (pessoa e tempo) antes de fechar a contratação (tira tempo de desenvolvimento e pode ter custo com um projeto que não vai ser realizado)

09 - Analog 01

Interview – 21/09/21

Em geral, um projeto novo começa com um conjunto de especificações (documento, email ou reunião), as vezes com arquitetura definida se não já começamos a montar os circuitos. Com isso já vai **aparecendo necessidades de controle digital** e vamos esquematizando uns diagramas de sinais ou algo assim do que vai precisar para controlar o bloco, trimming, etc. Isso é passado pro digital.

Uma coisa que é muito importante é definir como o **chip será ligado**. Já aconteceu de ter problemas no power up porque **não definiu direito o controle, não simulou certinho**... geralmente conseguimos remediar, mas temos nos atentado mais a isso...

Na integração com o digital, não tenho experiência com simulação mixed, mas tento acompanhar e revisar.

Se é um circuito novo, eu gosto de criar um modelo de pelo menos parte do circuito (para entender melhor ele). Depois, indo pro esquemático, as vezes crio um modelo VerilogA para gerar o controle do meu bloco e eu poder simular (dependendo da complexidade). Depois é integrado no digital mesmo na simulação AMS.

O controle necessário as vezes é discutido com o digital para ver a viabilidade.

Depois tem a **integração física**, onde tem a entrega do layout e os **problemas que (até com frequência) vem quando integramos o digital**, que é **sempre ali no final perto de tapeout**. Um problema muito frequente é subvias... também **problemas de DRC**, tanto que precisam ser resolvidas quanto problemas que não são para ser resolvidos: você tá lá fazendo o layout na fase final e recebe aquele bloco digital com **vários erros de DRC (mesmo que sejam para dar waivers)**. Aqueles (milhares) de **erros entram no fluxo e você se perde**... tendo que **resolver isso nos ultimos dias**.

Já discutimos isso em lessons learned, tivemos uma discussão com o digital de achar uma forma para **adiantar o mapeamento desses problemas** (seja um pre-tapeout com o digital) para quando chegar uma entrega perto do final já ter boa parte dos problemas mapeados.

Definição de interface e controle geralmente não dá muito problema. As vezes um pouco de **problema de entendimento: passamos a especificação errada ou incompleta**. Mas depois tem umas interações e vai resolvendo. Nós do analog temos que **aprender como passar melhor essa especificação**, essa informação... acho que ajudaria a antecipar problemas.

Não lembro de nenhum caso que o digital pediu algo pra mim.

De visão de topo, eu fui adquirindo um melhor entendimento do topo. Em projetos recentes, temos uma boa visão do nosso topo e um pouco do que tem de fora (nos envolvendo com quem vai cuidar do topo "final").

Em geral, o **topo é uma coisa que fica pro final** (tanto da parte AMS quanto analog). A parte de power up é um exemplo de simulação que deixamos pro final e que poderia ser feito antes (mesmo sem o digital). A parte da AMS também (digital conversando com o analógico), acabamos tendo **só uma verificação nos ultimos dias**, não sendo uma coisa que é estressada ao longo do desenvolvimento (por ser **muito em cima da hora**, acaba sendo uma **verificação não completa**).

Isso de **deixar pro final é sempre porque o projeto é muito corrido**. É difícil, mas eu entendo que é

complicado de lidar. Se tivesse alguém designado para cuidar dessas coisas.

Sobre impactos negativos no projeto, a questão das horas extras é uma coisa bem importante. Um projeto muito atropelado causa uma quantidade de trabalho muito grande no final e acaba fazendo os projetistas trabalhando muitas horas a mais e muitos dias seguidos, deixando a gente muito cansado e mais chance de fazer um erro. Eu acho que isso é uma coisa bem crítica. Entendo que tem um prazo ali e precisa dar um gás a mais, mas aos poucos tem que se achar uma solução para evitar isso.

Já aconteceu problemas de valor default errado de registrador (porque foi comunicado ou feito pelo digital errado), power up e down, coisas assim... bug grandes de comprometer o circuito não acontecem. Na bancada dá para dar um jeito e medir se precisar.

Eu me cobro bastante e fico frustrado quando tem algo que eu podia ter corrigido melhor ou não viu em simulação, mas coisas pontuais. Em geral, fico estressado com esses longos dias com frequência mas são coisas momentâneas na semana do tapeout.

10 - Manager 03

Interview – 22/09/21

Antes do projeto, tem uma parte de cotação e lá se estima um período/esforço para os times analógico e digital. Os projetos que participei eram grande parte analógicos e o digital auxiliava, então em algum momento rolava uma especificação. Em um projeto mais antigo, fizemos uma descrição em time diagrams de 3 cenários de acordo com as entradas. Esses desenhos foram usados pelo digital gerou um documento para usar como especificação. Em outro projeto mais novo, teve uma descrição quase que textual porque era um bloco digital mais simples.

Sempre há uma dor de cabeça no início do projeto para estabelecer o fluxo (standard lib, PDK, ferramentas).

No desenvolvimento, tem se consolidado a interação de layout.

Dos projetos que participei, não teve tanto simulações mixed signal antes de uma fase avançada. O analógico acaba fazendo (se necessario) um modelo VerilogA próprio que mais ao final do projeto era trocado com o digital. Em casos de apenas setar constantes, o próprio testbench cuida disso.

Eu acho que a interação com digital não faz parte do desenvolvimento analógico... o mixed signal fica no final (70% do projeto) para verificar que a comunicação tá boa, que os dois lados estão fazendo a coisa correta. Em nenhum momento o digital vai fazer o RTL no início para o analógico pegar e ir usando.

Na parte de layout, rola essa troca de abstract. Acho que não tem uma ordem certa (digital -> analógico ou analógico -> digital)... tem uma interação para definir o abstract. No layout, aquele abstract fica esperando para ser atualizado (trocado) por novos releases do digital. A simulação mixed signal vai se baseando nos releases do digital (RTL, prelayout, postlayout).

Em algum momento, tem que ser feito uma simulação de power up com o esquemático total (inclusive digital).

A simulação mixed hoje não é para validar a performance do chip, é só para validar que os dois lados estão conversando.

A visão de topo eu acho que o analógico considera o digital mais um blackbox. Eu acho que é entendido por quem mexe no topo, entendem a função do digital naquele sistema. Quem mexe com os blocos não é cobrado para ter esse entendimento, acaba dependendo do interesse de cada um.

Se todos os subblocos (digitais e analógicos) estiverem bem especificados, teoricamente não precisa ter essa visão de topo. Eu acho bom ter pois as pessoas podem ter outras ideias, mais gente para criticar. É bom, mas não acho necessário se tudo estiver bem especificado.

Quanto ao digital, geralmente é feita uma encomenda a eles. Eu não tenho penetração no time digital, eu não discuto como é feito o digital. Não sei dizer se o digital tem essa visão de topo, mas minha impressão é que não, não parecem ter interesse no topo como um todo. Em um projeto teve uma especificação melhor e talvez não precisasse saber mais do topo.

Na parte gerencial, também não tenho controle sobre gasto de horas do digital. O gerente digital cuida desse lado, aloca mais pessoas para fechar no prazo e acaba estourando o budget do projeto. Um projeto com tecnologia nova foi aproveitado para testar fluxo, testar ferramenta... acabou estourando muito o gasto do projeto com horas usadas sem estar relacionadas a entrega do projeto.

Um problema é a interface de armazenamento entre times. O analógico usa a base de dados das ferramentas (e SOS) e o digital usa o SVN. Não existe um consenso de como fazer essa interface. Já aconteceu de as configurações (SOS, libs apontadas) de dois projetistas analógicos estarem iguais mas estarem vendo versões diferentes. Isso acontece porque um tá com a cópia local do SVN desatualizada, um tá apontando pra trunk do digital, outro para uma tag... Fica a merce do projetista que tá abrindo o sistema. O que eu tenho feito é centralizar em mim: faco o checkout (local) e coloco no SOS as definições para apontar para essa cópia, aí todos usam isso e ajudou a corrigir essa questão de todos apontando para a mesma entrega. Tem dado certo essa estratégia.

Outro problema é a questão da lib de tecnologia digital que o time refaz devido as vias. O topo referencia o PDK e, ao instanciar algo da lib digital e exportar, as vias sumiam por causa dessa incompatibilidade das libs da tecnologia. A gente resolve porque temos na cabeça que isso pode ser um problema mas não é robusto. Conseguimos contornar fazendo o topo referenciar a tecnologia digital, mas talvez existam casos que podem causar problemas.

Outro problema de entregável, tem sido pedido coisas digitais, como modelo Verilog de bloco analog. Num projeto, tem sido feito um Verilog dos subblocos para fazer o topo. Tem sido interessante porque clientes podem pedir isso para o fluxo digital e pode acontecer dificuldades pois não é o domínio do pessoal analógico. Outro projeto também tinha entregáveis digitais tipo LEF, LIB, Verilog, CDL... Alguns questões do LEF (cell type "block", definição dos pinos como "power" ou "signal", sentido "in" ou "inout") que não são problemas pro analógico mas afetam o digital (essas definições não estão estabelecidas, falta um acordo).

A lib OA do digital nunca se sabe muito bem o quão completa está.

11 - Analog 02

Interview – 23/09/21

O fluxo que geralmente é feito é que tem que entregar rápido as coisas, então o ideal nunca é feito. O fluxo mais adequado geralmente seria o fluxo de top-down: levantar spec do sistema inteiro, criar arquitetura, especificar subblocks (modelo, interface, etc) e implementar de baixo pra cima, substituindo aos poucos no topo.

O que geralmente é feito é que os blocos já são feitos em esquemático de primeira (bottom-up) que é adequado em alguns projetos, mas tem que tomar cuidado se a arquitetura é muito grande ou tem coisas nova e tal. A maioria dos projetos que participei montava uma arquitetura e já ia desenvolvendo a nível de transistor todos os subbloco. Bloco a bloco não tem muito problema, mas para a integração é pessimo porque não dá para prever os problemas (demora pros blocos ficarem pronto, simulação demorada). Eu acho que só trás coisa ruim pro projeto como um todo.

Em um projeto, teve um ponto bem positivo de comunicação entre times porque foi criada uma especificação bem completa usando diagramas de tempo pro digital. Facilitou a comunicação porque ele já tinha definido tudo e foi criado um modelo para ser usado enquanto o digital trabalhava. Foi muito bom pro projeto em geral.

Outro projeto também teve especificações de máquina de estados, um fluxograma para o time digital. Ajudou muito a comunicação. Em outros projetos, faltou saber como seria a comunicação com o time digital, pois não se sabia se o digital ia poder fazer, se outra pessoa ia fazer... passava tempo, ninguém do digital podia, aí tinha que fazer o modelo por conta própria.

A parte da verificação do bloco digital é desconhecido pro analog (não se sabe o que foi verificado), mas é gerado um bloco, verificado e criado uma tag no SVN. Nós recebemos o caminho do arquivo (seja RTL, prelayout ou postlayout) para simular no analog, validando incrementalmente. As constraints do bloco são diferentes do que analisamos no mundo analógico, mas também são válidas e tem que ser cheçadas.

Uma coisa que falta, com certeza absoluta, é planejamento. Se vai ter participação do digital, o que eles vão fazer, quando vai ser feito e como vai ser feito. Falta reuniões no inicio do projeto para alinhar esses pontos (mesmo que mude depois). Se houver, organizar isso. Não fica claro se vai ter gente pra ajudar.

Algo que peca muito é a comunicação entre os times para acompanhamento. Poderia ter reuniões mais formais além dos acompanhamentos por ferramentas de comunicação e gerencia. Apesar disso, a comunicação é boa e as coisas andam bem. As vezes falta um pouco de transparencia e sincronia. A comunicação até é boa, mas a organização faz a comunicação não ficar tão boa quando poderia.

Com certeza absoluta falta uma visão de topo. Em muitos projetos existe uma caixa preta enorme que muita gente não sabe porquê está fazendo algumas coisas. Essa é a importancia da reunião inicial de planejamento (que seria o kick-off) para dar uma ideia geral do que era pra ser feito e definição de responsabilidades. Já vi muitas vezes um projetista receber uma demanda que é uma caixa preta que ele tem que fazer. As vezes tem problemas de interface que se ele soubesse ele poderia mudar no bloco dele, isso não fica tão claro.

A falta de visão de topo atrapalha o projeto como um todo. Se os projetistas não sabem o motivo do seu bloco ou com quem o bloco vai conversar, isso dificulta muito o entendimento. Quando tem uma integração, ao inves de várias pessoas que podem contribuir, fica uma pessoa que tem que

cuidar de vários filhos que não entendem o todo. Não precisa ser algo complexo, mas todo mundo tem que entender o sistema e deixar tudo bem claro no início do projeto. Até quando for debugar algo, todo mundo pode ter uma referência do que pode ser feito, do que pode ser analisado. Um exemplo de um projeto que participei da integração, tinha vários blocos que eu não sabia o que tava acontecendo e tinha muita conversa minha para entender o que faziam. Ou o contrário: muita conversa dos projetistas comigo para entender algo do topo. Falta um pouco dessa conversa entre o time.

Algo que sempre vejo problema é a questão da atualização das informações. Quando eu espero uma entrega, acho que o mais adequado seria um documento claro com o que tem que ser feito / o que tem que ser atingido (documento de especificações de bloco, porquê de algumas coisas), além de informações válidas: mudanças de projeto aconteceram, documento de spec não foi atualizado, datasheet não foi atualizado... fica um conflito onde spec fala uma coisa, datasheet reporta outra e simulação que gera outro resultado. Depois tem um esforço de conferir e atualizar tudo com o que foi medido no silício. Esse conflito de informações é muito problemático e sempre acontece.

A integração e verificação de topo vem melhorando. O uso de uma pessoa focada em integração desde o início do projeto ajudou em vários projetos. Alguém dedicado para monitorar os problemas no começo faz o início do projeto ser muito mais rápido. O ponto que tá faltando é essa questão dessas reuniões de kick-off serem feitas com mais cuidado, deixando claro quem ficará responsável pelo topo.

As comunicações no geral são boas (clara e objetiva) nessa questão. É a organização que complica mesmo, talvez um documento ou outro para facilitar a vida, um padrão ou outro a ser seguido. Independente disso, quando é preciso se trocar informação, a troca é boa.

Impactos negativos nos projetos geralmente são causados por causa da correria. Minha impressão é que os prazos são sempre apertados demais, tudo é feito na correria para atingir o prazo, muitos bugs ficam pra trás ou não são encontrados (avaliação não foi tão cuidadosa). Se o prazo fosse um pouco maior e o planejamento melhor (usando checklists para garantir que foi feito o que precisa para dar certo), reduziria o número de problemas.

Os bugs que passam não são graves... geralmente. Em um projeto, um bug passou e pode impedir o chip de ligar. Um sinal em um determinado domínio de tensão não tinha level shifter, o que pode queimar o sistema. Uma coisa muito simples que passou. Se tivesse sido feita uma verificação mais cuidadosa, não teria passado. Muito probleminha de interface... muita coisa simples e que passou despercebida.

Eu fico um pouco frustrado porque são coisas mapeáveis e resolvíveis, que já foram ditas várias vezes e soluções levantadas nunca foram implementadas (existe um engessamento em algumas coisas). Quando é um problema grave em uma situação muito específica, acontece, mas problemas simples que um teste básico teria pego é complicado. O resto do estresse considero aceitável.

Para finalizar, acho que essa questão da organização que falta ser melhorada. Organização do fluxo de projeto... ou talvez respeito a ele. O kick-off existe, mas é raramente feita. Essa definição das atividades também precisa ser organizada. A empresa tem capacidade de fazer projetos bons e em pouco tempo, só falta planejar melhor e seguir o plano.

Extra

localizacao dos documentos é um problema

documento sempre falta do lado do analogico (prazo pequeno, deixa documentacao de lado, depois tem que rodar tudo de novo para documentar de novo, e perde informacao)

cell id exist, tem uma estrutura mas ninguem segue (mesmo estando na wiki a estrutura e não detalhe as seções, muito superficial) Até entender o que tem que preencher demora muito.

Cell id as vezes nem é mais pedido, foca no datasheet. Muito projeto tá só com datasheet, pois o cellid era pra ser um historico de tudo, mas é muito grande... é um bom documento, tem uma estrutura bem completa... foi pensado em ser feito durante o projeto e faz sentido, mas era esperado que durante o desenvolvimento o cara ia preenchendo as seções e isso nunca é feito.

é bom o projetista ter flexibilidade, mas é importante ter pontos obrigatórios que tem que ser seguido e bem estruturado. Tem que ter cobrança e seguir o fluxo

Podia ter um poster gigante indicando onde cada documento importante da chipus está.

Tem coisas cruciais que não são feitas... teve um projeto que não foi feito simulacao de chip ligando (power up).

12 - Manager 04

Interview – 27/09/21

Tudo começa quando recebemos alguma especificação do cliente (externo ou não). Ai já diferencia do que deveria ser: **as specs nunca estão completas**. Elas sempre chega com **alguns requisitos muito bem definidos, alguns com entendimento parcial do que vai precisar e outros sem nenhuma definição** (discutimos com o cliente para definir essa feature).

Entra a especificação, o gerente faz um breakdown baseada nela, faz uma estimativa de esforço/cronograma e divide as tarefas entre os projetistas. Projetistas recebem o mesmo nível de especificações que a gente, que **sempre estão incompletas**... incompletas não no sentido maldoso, as vezes o **cliente não tem resposta para tudo e temos que descobrir enquanto estamos fazendo**... não é porque não temos feito um trabalho de captura de spec bom, porque não é discutido, porque não tem se tentado obter as informações antes de começar o projeto... mas em geral **entre discutir e começar o projeto o cliente não tem todas as respostas**.

Da parte da **especificação do digital**, pode chegar **do cliente (se já estiver bem definido)**. Muitas vezes vem uma **especificação geral do chip e o gerente tem que bolar a especificação do digital a partir dela**. Essa etapa é um questão antiga... as vezes **passamos o como funciona, sem ter um formalismo na especificação**. Isso acaba gerando **muito retrabalho lá na frente**... pois eu sinto que nós recebemos uma especificação parcial do cliente e enviamos pro digital ainda mais parcial, um esboço grosseiro do que tem que ser.

Vamos **levantando e refinando os requisitos a medida que a parte analógica vai sendo projetado**, vamos tendo um entendimento do que se precisa do controle digital... Vai **aprimorando e passando aos poucos**.

Essas **especificações incrementais são absolutamente pertinentes** porque você **não tem uma visibilidade do que vai ser necessário**, acabamos precisando já começar a trabalhar para não deixar tudo em cima da hora, fazer as coisas de forma incremental (apesar de **não termos uma dinâmica bem estabelecida do que é esse incremental**). Porém algumas vezes entra na correria e os projetistas analógicos apenas passam uma ideia do que precisa ser feito (depois vai aprimorando), mesmo sendo capaz de gerar uma especificação formal, mais detalhada.

O principal problema das especificação é que a **informação geralmente vem em cima da hora e acaba dando correria para rodar todo o fluxo**. O grande problema tem sido o **timing da relação entre os times**. Até hoje, eu considero que os blocos que pedimos pro digital não são complexos então eu acho que o grande problema são esses **requisitos em cima da hora**.

Essas especificações em cima da hora ou incompletas é porque **não sabemos todas as features que vamos precisar do digital, todos os sinais de controle ou valor default dos registros**. Como sabemos que partes do fluxo do digital são automatizados, não tem muita complexidade técnicas... **mas dado em cima da hora, até rodar o fluxo todo gera correria**. E aí tá o problema.

Acho que a documentação também fica comprometida por passarmos as especificações dessa forma.

Acho que existe uma lacuna quando falamos de comunicação saindo do digital e indo pro analógico. Até hoje, **se eu for fazer um pedido, eu não saberia quais documentos preencher e como são preenchidos para passar essa especificação pra vocês**. Acabava que vocês preenchiam e nós revisavamos.

O digital pedir informação pro analog acontece, documento não.

Do lado analógico, após essas especificações, tem simulação, corner, fechando spec... **Em algum momento** fazemos simulações mixed signal quando começa a voltar informações do lado digital. Temos tido problemas no fluxo devido a ferramentas. Começamos as simulações usando RTL inicialmente e, mais pro final, deveríamos fazer as simulações com tudo em device-level. No entanto, não sei se temos conseguido fazer essa parte ou usado SDF para anotar timing do digital. **Em algum momento**, temos feito algumas simulações de power up, depois vem sign-off...

As **simulações mixed-signal tem problemas técnicos** (definir interface elétrica, connect rules, dominio de alimentação). O que tenho visto é **desentendimento do que uma feature deveria fazer, ou coisas simples como polaridade de um sinal, ou quando um sinal tem que ligar/desligar, em quais condições deveria funcionar**. Versionamento era um grande problema antes, sei que melhorou e não acho que tem sido um problema.

Tá aí uma coisa que estamos tentando fechar no fluxo mixed... **como saber se a verificação que tá vindo do digital é a mesma que deveria estar rodando no topo**. Aquela coisa de validar modelo, validar verificação de um lado para quando rodar no outro ser a mesma coisa que tava esperando.

A maior dificuldade é técnica no mixed, mas eu acho que tem uma **dificuldade em como fazer o fluxo de informação da especificação rodar de uma forma sistemática**. Ainda é muito baseada em conversa, email e reunião... ainda **não temos um fluxo estruturado para passar especificação pro digital** e, de alguma forma, a documentação voltar para existir uma revisão do que o digital entendeu para ver se já se identifica de cara algo que não foi entendida corretamente.

Esse fluxo de especificação acaba causando coisas como uma **feature dando problema e é pega por uma simulação que nem era para pegar aquele problema**. Tem um **problema dessa troca de informação sobre especificação**.

Os **entregáveis da parte física tem tido problemas**. Aquela **ideia de que é só colocar o bloco digital e DRC/LVS vão passar não tem sido bem assim**. Eu tenho tentado conversar bastante para melhorar isso aí, mas **sempre tem alguma coisa manual que o analógico tem que fazer ou muita interação com o time de backend digital para fechar o topo**, conseguir rodar tudo (antena, DRC, LVS). Por fim tem o sign-off com checklists para fechar tudo.

Na integração e layout eu não sei detalhes técnicos, mas **é evidente que a entrega do digital vem com problemas, gerando problemas ao integrar**. Mesmo você tendo um abstract, tendo feito um LVS e DRC com blackbox com uma entrega intermediária do digital... **quando importa o GDS geralmente tem problema**. Agora se o problema é antena (não teve essa especificação de proteção para um sinal específico), ou porque não bate LVS... **ainda não está plug'n'play**.

Não existe algo sistemático na Chipus para que todos tenham uma visão de topo. Apenas quem está envolvido diretamente com o topo ou curiosidade do projetista. Acontece muito do projetista só enxergar o seu bloco. Acho que **não existe uma regra geral se isso é bom ou não...** seria legal todos terem para ajudar ou propor soluções... **seria um problema se o cara não tiver nem uma visão do macro bloco onde seu bloco vai ser usado**. Não ter uma visão geral de um chip complexo não vejo ter um impacto na qualidade do que estamos fazendo, mas não saber o macro bloco em que está inserido sim. Não sei a frequencia disso.

Problemas de integração, o que acontece as vezes é uma questão de **como é gerado a spec de subblocos**. As vezes é teórico (planejado), as vezes foi feita uma simulação comportamental para gerar a spec... quando você **traz o bloco e a interação entre os blocos não funciona ok** (carga maior, sinal devagar, kick-back, etc)... **Acontece sim, mas não sei se é porque o cara não conhece o topo...** pode ser que sim, porque se o cara recebeu uma spec com carga X e antes de desenhar ver onde o bloco estará inserido pode ver que a carga não bate.

A comunicação entre os times é sempre bastante positiva, nossos times estão sempre tentando

resolver problemas e trabalhar com o que tem. Linguagem não acho ser um problema sistemático, se aconteceu são coisas pontuais. As partes **conhecendo suficientemente bem o trabalho um do outro** também para tentar ajudar nessa parte da conversa (dicas de como fazer).

Temos intensificado o uso de ferramentas para registro e centralizar as informações. Nos antigos projetos **aconteciam demais de perda de informação (porque ou como algo foi feito)**, pois não tínhamos uma **cultura de documentar passos, tomadas de decisões**. Acho que ainda existe (pode estar documentando em algum lugar), mas a **rastreabilidade disso ainda não é simples** (tem que procurar muito para achar). Acho que existe um fator de erro humano, as vezes acontece da pessoa não registrar, não dar atenção ao issue... Teve um caso muito atual que um pedido do cliente ficou registrado em ata e não foi feito... A entrega foi feita sem isso. Ainda não existe aquela cultura ou uma preocupação sistemática de que, se tem algo na ferramenta é porque teve um pedido para se resolver um problema... antes de ir para tapeout, tem que ser feito ou dado waiver para todos os items... isso não acontece ou não é registrado, é um problema.

É uma etapa de fechamento que estamos precisando melhorar. A própria **checklist de tapeout que comentei ainda não tá na ISO, alguns gerentes não tem dado prioridade por causa de tempo ou outras atividades** (é para ser realizado em momento crítico do projeto).

As atividades de **integração e verificação de topo** tem sido feitas mais **pro final** dos projetos. Temos tentando lutar contra isso com a figura de um **responsável pela integração desde o início do projeto**. Temos tido dificuldade porque se aparece algum incêndio essa pessoa é mudada para resolver problemas de bottom level.

Verificação de topo acaba sendo feita no final mais para caçar bugs do que propor melhorias ou alguém olhando para as especificações dos blocos para antecipar falhas de interação dos blocos... Acredito que temos feito pouco nisso pois noto que essas atividades continuam centralizadas no final. E tenho visto **muito essa dificuldade de fechar essa questão de garantir que o que tá sendo verificado no fluxo digital é o mesmo que vai ser verificado no mixed-signal**. Ainda não temos uma solução para isso, quem dirá virar algo sistemático.

Os maiores problemas devido a integração seriam **bugs tardios ou bugs não encontrados**. No fluxo mixed signal não temos a mesma cobertura do digital, então acabamos simulando só as condições específicas, mais críticas... bugs não críticos mas de alguma relevância são pegos tardiamente ou não são pegos.

Os impactos negativos em projeto são **custo com certeza** (gastamos mais semanas a mais do que o planejado devido a **retrabalhos ou trabalhos muito desestruturados** no final do projeto por causa da correria). **Cronograma também**, com duas facetas: no caso em que o projeto **estoura o prazo** é negociado com o cliente, nos casos em que isso não é possível (MPW ou data inegociável) acaba gerando **muitas horas extras** (influencia indiretamente no custo e impacto humano, causando impacto em outros projetos). **Funcionalidade** geralmente relacionado a bugs não encontrados ou, no caso de bugs encontrados tardiamente que sejam complicados, a correção não é ideal, as vezes até tendo que negociar requisitos com o cliente, impactando a funcionalidade de alguma feature.

Eu acho que o **estresse na época de entregas** é esperado, mas muitas vezes é **acima do aceitável**. Um lado é a **preocupação que você tem do projeto dar certo e que você fez tudo que podia**. Muitas coisas não estão no seu controle, tem que **confiar nas outras pessoas, que pode causar mais preocupação** principalmente se for alguém ou uma situação nova, gera um pouco de **ansiedade e insegurança**. Correria, checklists que não foram feitas a tempo, horas extras... Isso gera um estresse acima do que deveria... Ver os projetistas tendo que enfrentar essas coisas e pedir para trabalhar no final de semana, horas extras... também me gera estresse e frustração.

Nossos gerentes metem a mão na massa pesado. Entra na correria, eles passam madrugada, não deixam a peteca cair.

Uma coisa no fluxo que acho que **estamos devendo são os design reviews**. Não sei como tem sido

feito no digital, mas no analógico geralmente isso é pulado e o review acaba acontecendo nas reuniões ordinárias. Geralmente discussões com o líder de projeto até as questões estarem entendidas. Não tem aquele momento formal de parar, fazer um design review, trazer outras pessoas com visões diferentes do bloco. Ele é previsto no fluxo, mas não temos feito com a frequência que deveria.

13 - Manager 05

Interview – 27/09/21

O fluxo de projeto varia um pouco... o que eu vejo é que em alguns projetos chega um problema e sugerimos uma solução. Temos a concepção do esquemático pelo time analógico, no meio do projeto quando algumas especificações já estão mais bem definidas entra o digital. Digital recebe os inputs necessários do analógicos e o layout providencia uma área, um shape específico...

Pro layout, fazemos uma abordagem híbrida... começa top-down, continua bottom-up e depois tem uma convergência no meio. Nessa convergência no meio, sobra uma área pro digital e os blocos analógicos já estão todos posicionados e a pinagem está posicionada e conectada com buses já na ordem correta pro digital ajustar na ordem que precisa.

Originalmente, pegávamos um espaço do layout, desenhávamos um abstract e enviávamos pro digital. Mas como deve ser esse abstract? Que layers fazem sentido pro fluxo digital? Pitch, posicionamento, níveis de metal, orientação, tamanho... Essas coisas. Muitas vezes entregava um abstract e a entrega em OA lib estava totalmente diferente. Isso acontecia porque meu input não servia, porém eu não era avisado disso.

Rolava aquele estresse porque o digital fica sempre pronto no final do projeto, mas dá problema de LVS, memória,... Ainda tem isso, ainda sofremos disso... Parece que as verificações físicas do fluxo analógico são diferentes do digital, sempre causando essas dificuldades.

No layout, sofremos um pouco por causa do digital... tanto que um projetista está aprendendo todo o fluxo digital para entender as ferramentas e processo, com o objetivo de tornar mais suave essa interação layout-digital... para quando o digital pedir algo (às vezes não muito claro) conseguirmos entender melhor o pedido e a necessidade para entregar o que faz mais sentido.

Essa interação com o digital tem melhorado muito porque melhoramos a qualidade das entradas de layout e o pessoal do digital tem se flexibilizado mais (alterar scripts, ajustar orientações de metal, por exemplo). Essa troca tá longe de ser perfeita (acho que teria que ser muito burocrática... formulários, padrões, etc), mas nós já melhoramos muito (SVN para pegar OA lib, enviar abstracts atualizados, etc).

Ainda é estranho pro analógico as noções de pitch, track, layers usadas,... Por isso um layoutista pegou o Welcome Guide do digital, fez o tutorial e gerou um documento voltado pro time de layout com esse fluxo. Também explorou como um layout com shape não quadrado é feito, para entendermos o que precisamos fornecer para a ferramenta conseguir fazer isso. Pessoal do digital está envolvido pois tem coisas triviais pro digital que não é pra gente.

Essa iniciativa do layoutista aprender o fluxo digital é porque o digital (assim como todos na Chipus) estão constantemente ocupados. Blocos pequenos (como vários bin2bubble) sempre são necessários e não podemos esperar muito tempo para alguém do digital ficar disponível para fazer isso pra gente. São coisas simples, então para essas coisas mais básicas o layout vai aprender a fazer. Isso vai herdar um conhecimento muito legal de como tem que ser o input de forma a reduzir o ruído entre os times.

Comunicação tem problemas de ambos os lados, isso porque é sempre aquela comunicação rápida (requisição pouco detalhada). É muito difícil quando o digital dá um grau de liberdade pro layout... "Faz do jeito que quiser que eu dou um jeito e a gente vê se dá". Eu vou usar qualquer pino, qualquer layer, qualquer coisa em qualquer área e entrego. Acaba gerando problemas como a ferramenta digital não reconhece aquilo como pino, tracks/grid que mudam orientação dos layers...

São informações que não são passadas logo de cara.

Nunca recebi um input do tipo "faça um abstract desse jeito, partindo da borda na horizontal um track de 0.56 com pino de tamanho de 0.28".

Em um projeto, depois de muitas e muitas interações tentando entender o motivo do digital vir errado, teve uma explicação do porque a ferramenta digital fazia isso. Dai clareia, mas nesse ponto já está na etapa de exportar GDS para o tapeout, com nervos a flor da pele. Se issosof já estivesse claro desde o início, iria reduzir uma quantidade de interações muito grande.

Na etapa final de layout do digital, geralmente dá problema de verificação físicas (especialmente LVS). O digital libera o layout 1 dia antes do tapeout, mas estava usando uma suite de LVS diferente da ferramenta analog (com uma customização que não faz sentido para o analógico por exemplo). Acho que se o digital fizer o layout com as regras e verificação física que o time analógico usa, talvez um dos problemas desse pra resolver. As vezes tem que ser resolvido pelo próprio time de layout.

O digital faz um entregável que pra ele funciona, mas pro analógico não.

Domínio de alimentação do digital também as vezes gera problema. Como geralmente eles tem apenas um domínio, pode acontecer curto no substrato quando vai pro topo que não é um problema para eles. Essas acordancias não são feitas antes para definir as tensões do substrato. As vezes é fácil de resolver, as vezes não. Mas comparado a outros problemas, pode até ser desconsiderado.

Uma coisa que sempre nos atrapalha e o digital precisa nos ajudar é passar LVS numa memória ou no digital com um IP de um terceiro. "Tem que incluir o CDL e as standard cells dessa memória". Ficamos perdidos, acabamos perdendo um tempo antes de pedirmos ajuda... Poderia ter um documento bem simples so explicando isso e indicando os path dos arquivos. Nunca tem isso.

Também sinto que as vezes falta comunicação para o time de layout analógicos. O crescimento de um bloco digital as vezes fica entre design analógico e digital, ou uma modificação do RTL causa um aumento. As vezes só descobrimos quando chega a entrega já com o aumento.

Acho que tem interesse da visão de diferentes times, mas os times digital e analógico fazem layout de forma muito diferente, com ferramentas diferentes... Pro analógico o backend digital é obscuro, onde ficamos em busca de informação para clarear e podermos contribuir. E vejo isso no digital também, pois perguntam muito.

As vezes vejo que um time fica sofrendo fazendo algo que é extremamente simples para o outro time, como chop/cut/strech pro analógico ou rodar os scripts do digital. Essa falta de visibilidade de um lado ou outro acaba causando isso. Temos tentado interagir com o digital para trocar informações.

A visão de topo entre digital e analógico geralmente tá claro. Principalmente quem estiver envolvido com integração. Eu acho que não tem sido um problema ultimamente.

O maior problema da integração é o prazo tardio que recebemos. Geralmente porque o digital tá muito ocupado ou com mudanças de requisitos em cima da hora do analógico pro digital.

Questões em circuitos grandes, como correção de antena, as vezes nem passamos pro digital porque conseguimos falar ele funcionar então tentamos resolver pro fora para não alterar o digital.

A comunicação entre os times em geral é boa. A comunicação fica centralizada no gerente, porque tem muita troca dos projetistas digitais... pra gente fica meio bagunçado e acabamos falando direto com o gerente.

Da perspective do layout, o que está implementado lá dentro não importa. Se passar DRC/LVS e

coube na área, fechou. As vezes conseguimos compartilhar informações de quanto o sinal vai percorrer um longo caminho ou algo assim por questões de fanin ou fanout.

O impacto negativo que vejo é stress. Em projetos menos organizados, a saúde mental foi muito afetada. Em poucos projetos mais atuais, tivemos níveis aceitáveis. Na correria do final do projeto pode ter atritos entre colaboradores devido ao stress.

O que a gente pode fazer? Vamos interagir um pouco mais? Para quando fizermos uma pergunta o cara entenda o que estamos falando e a gente consiga pedir ajuda de forma mais clara e objetiva, podendo digerir o que foi falado.

14 - Analog 03

Interview – 28/09/21

Recebemos o pedido do cliente, o tamanho é definido pelo esquemático e o shape conversamos. As vezes o cliente especifica a região dos pinos ou outras detalhes (seja interno ou externo). O esquemático inicial precisa ser cedo, nem que seja uma noção para conseguirmos começar o layout. Precisa estar minimamente maduro, não necessariamente completo. Tem casos em que **fazemos uma simulação extraída e mudanças são necessárias** (nossas ou no esquemático), ou o esquemático ainda está sendo desenvolvido, mas a topologia tá meio que pronta.

Começamos com um floorplan de todo o sistema, delimitamos uma área pro digital e foi verificado com o digital se era suficiente. Foi pedido um abstract (tamanho, shape e posição estimada dos pinos) e um layout foi criado a partir daí. Fizemos ajustes no tamanho e não ficou exatamente igual porque tem aquela questão dos tracks e tal, mas a interação foi boa. A primeira entrega ao cliente já tinha o analógico pronto (ou semi-pronto) e um abstract no lugar do digital devido a dificuldades no digital e na integração.

A primeira entrega que recebi do digital foi um esquemático só com os pinos. Também me passou também uns CDLs das standard cells e da memória. **Tentei exportar o esquemático de tudo e não deu certo**. Ele me passou um modelo que **para ele tava funcionando mas pra mim não**. Depois ele criou uma view esquemático com todas as standards cells e tudo mais, não dava para entender mas deu certo (ainda teve que colocar CDL das stdcells). O que mais complicou a integração foi **entregar algo que fosse usável pra gente**. Não especificamos o que era entregável pra gente e também não temos padrão de entregável do digital pro analógico.

Na parte do abstract foi bem tranquila. Teve alguns momentos que o pino tava em posição errada, mas fomos ajustando isso. Um pequeno problema é essa **variação da posição de definimos** devido as tracks. Fica meio desalinhado mas não é muito grave. A integração foi bem legal, mas eu **não tinha muita noção do fluxo digital e estranhei quando notei essa diferença**. Me explicaram, **tinha poucos sinais e não tinha muito problema de roteamento nesse projeto**, então foi de boa.

Nesse projeto meio antigo, o abstract era pego direto pelo digital da biblioteca de projeto, no caminho que informei.

Um projeto um pouco antigo teve uns **problemas de DRC/LVS relacionados a uma memória**. **Sempre que tem coisa AMS é trabalhoso**. Digital **verifica o layout de forma bem diferente**, usam outros inputs de esquemático (modelo, netlist, CDL) e **integrar isso na ferramenta analógica é mais complicado** (não consegue verificar o digital, o esquemático do digital depende de outros arquivo, usa arquivos que a ferramenta não entende, port order do CDF parma).

Port order já tive problema e já vi outros tendo também. Tivemos que gerar simbolo novo e arrumar na mão na view layout.

Faltou uma integração melhor entre os times. O analógico ficou pronto muito mais rápidos, ficou esperando o digital para entregar pro cliente.

O problema de entregáveis que sejam usáveis é o maior eu acredito. Seria bom ter alguns padrões e métodos (adaptados para cada projeto) detalhando se vai ser entregue só um esquemático, ou se vai ser preciso incluir inputs adicionais. Para conseguirmos verificar e exportar o esquemático completo do IP.

Da forma que foi entregue o layout tivemos problemas, mas depois que fizeram essa OA lib, ficou

bem tranquilo. DRC tiveram alguns problemas, principalmente por causa da diferença de tensão, mas foi pequeno e bem particular. A maior dificuldade foi realmente o LVS, fazer reconhecer layout e esquemático digital. O digital teria que se adaptar a essa forma para conseguirmos verificar.

A visão de topo eu considero importante ter. Nada adianta fazer ou modificar um bloco se não se adaptar a onde for usado. No início do projeto as vezes é difícil ter uma visão completa. Acho que no geral o pessoal tem uma visão do topo ou pelo menos do seu macro bloco. Seria bom mostrar isso de uma forma mais deliberada, pois já teve momentos que acho que pessoal saia para fazer outro projeto ou algo assim e isso se perdia.

Na minha experiencia em projetos meio antigos, comunicação teve mas faltou definir melhor o que era esperado como entregável e prazos (bem comum atrasar o digital). Documentação (do outro time) acho que não ajuda muito porque os fluxos são muito diferentes.

Impactos negativos são principalmente hora extra e stress. No final do projeto com problemas de layout, fica com a pressão de entregar na data certa, trabalha horas extras para cumprir, trabalhar de forma corrida... frustração de não conseguir entregar um trabalho com muita qualidade. Como o layout é a ultima etapa de projeto, afeta a saúde mental. Acho que não chegou a inviabilizar um projeto, mas a qualidade ficou comprometida.

15 - Analog 04

Interview – 28/09/21

Geralmente começa com um design pronto (idealmente) para fazer layout dele. Geralmente eu não faço um layout do zero, tem acontecido mais eu fazer verificação física, corrigir DRC/LVS, modificações por causa de mudanças no design... A maior interação que tenho é com a equipe de design, principalmente ideias e perguntas de como deseja que seja feito o layout, qual casamento, freq de operação, nós críticos...

As vezes preciso interagir com o digital quando precisa ser integrado em um bloco analógico. As vezes a coisa não encaixa muito bem ou precisam de inputs, como o abstract (onde estão os pinos). Geramos um abstract vazio sem blockages, só para gerar as informações de pinos cuja posição facilitaria para integração de topo.

Já aconteceu de colocarmos pinos fora das tracks digitais (muitas nets) e depois o layout volta com os pinos em outras posições (deslocados um pouco ou em lugares completamente diferentes). Não sei exatamente o porque disso, acho que o maior suspeito seja o abstract enviado (não está alinhado com as tracks do digital). Acho que é o maior problema que pode acontecer nessa interface.

Uma informação do abstract que poderia vir (ou o analógico verificar) é grid/tracks de roteamento. Seria importante saber isso.

As vezes acontece de recebermos blocos digitais e, ao fazer a verificação usando ferramentas do analógico ou abrindo o layout, aparecem alguns problemas de DRC (layout teve que arrumar, mas eram coisas simples) e LVS. Também já aconteceu do bloco digital vir com square brackets "[]" ao invés de angle brackets "< >" e isso gerava erros de LVS. É simples e resolver, mas o bloco que recebemos gera falha.

Teve um projeto em que o PDK foi mudado (regras mais restritivas) e o digital continha um isolation ring externo. Houve dificuldade do lado digital para conformar as cells digitais com o iso ring que já existia. Acho que o maior problema é esse iso ring não vir direto do digital, pois ele é necessário pro bloco mas não fazia parte do layout.

O bloco digital tinha que ser o mais plug'n'play possível, pois não conseguimos alterar facilmente. Se dá problema de LVS, não tem o que olhar pois não sabemos qual é o source do circuito, como era para estar... Sei que tem um esquemático gerado a partir do Verilog, mas não sei como isso é feito e é um esquemático bem difícil de enxergar.

DRC e LVS devem ser feitas no digital, mas no caso de LVS já teve alguns problemas causados por configuração de ferramenta ou port order diferentes entre os times. Foi mais um problema de integridade do que um problema do digital ou do analógico. Problemas de LVS não são recorrentes apesar disso.

Os blocos de topo que mexi já estavam feitos, só fiz verificação e ajustes. Acho que alguns blocos poderiam ter conexões melhores... cheguei a ver blocos que não estavam com uma boa visão do topo... por exemplo se um sinal está vindo de cima para baixo e o bloco tem o pino na parte de baixo. É possível que o topo nem existia quando esse bloco foi feito, talvez nem um floorplan de topo.

Em outro projeto mais novo (feito do zero), as pessoas se preocupavam com o topo mas na parte de executar não ficava tão bom quanto eu (topista) desejava. Eles tinham uma visão e eu que me ajustava para conformar como o bloco estava. As vezes é mais fácil arrumar no topo (mais espaço,

mais simples) do que alterar o bloco, mas havia discussão sobre esses pontos e balanceávamos as dificuldades. É importante ter essa flexibilização dos dois lados.

As entregas da Chipus tem uma coisa estranha... cada time usa uma ferramenta de versionamento diferente, tinha que ser tudo uma mesma coisa. Eu acho uma inconveniência. Para enviar abstract pro digital nós copiamos e colamos no SVN digital. O abstract volta como uma view layout no SVN que referenciamos para usar no nosso ambiente. Temos que dar update não só no SOS mas também no SVN. É uma pequena irritação para quem precisa integrar os dois mundos.

Essas trocas já causaram uns problemas por causa de tags. O gerente de projetos determina o path que devemos olhar para usar as coisas da digital e fica em um libdefs geral. Porém, já aconteceu de eu ter que colocar no meu libdefs local o path correto por causa de alterações no digital.

A comunicação muito boa, nunca tive problemas.

Geralmente atividades de topo são feitas mais pro final, gerando muita correria, trabalhar no final de semana... Trabalho se acumulava muito no final.

Acho que falta informação de coisas importantes. Em um projeto recente tivemos que perguntar coisas importantes após a entrega. Alterações de ultima hora quando estamos na correria e falha na obtenção de informações importantes para conclusão do projeto causaram atraso.

Ansiedade porque ficamos preocupados em saber se vamos conseguir entregar é o maior problema. Stress porque você fica batendo cabeça com a mesma coisa... você fica pior porque ta trabalhando muito, mas fica mais devagar e precisa trabalhar mais. Não trabalhei nos finais de semana, mas em um projeto tive que fazer horas extras por um longo periodo. Da loucura de entrega não tenho muito do que reclamar, mas bate ansiedade e stress.

16 - Analog 05

Interview – 29/09/21

Em projetos de integração do topo, tem a parte de consolidar as bibliotecas (Digital, IO, IPs) e definindo os IOs. Nesse ponto já temos uma ideia de como será o analógico, o digital... então já vamos montando um alto nível. Depois começam interações de interface de layout: definição e posicionamento de pino, floorplan e faço um abstract do digital. Antes o pessoal do digital se virava com a view, mas ultimamente tenho gerado LEF direto. As vezes volta um pouquinho diferente a posição do pino, mas coisa pouca.

Existem outros projetos que o digital faz subblocos auxiliares pro analog. As vezes são usados em blocos intermediários e são entregues muito tarde, fica tarde demais para receber, estima uma área muito maior do que precisava.

Nas discussões sobre interface muito no início do projeto, é definido algo e vai sendo incrementado ao longo do projeto. As vezes tem uma pequena diferença depois entre os projetistas, com sinais que não são mais usados... A atualização das mudanças não foi comunicado acaba gerando retrabalho depois.

O fluxo com os abstracts é uma solução. Antigamente dava uma especificação pro digital e só sabia como ia ficar quando tinha um layout pronto. A troca de abstracts foi uma solução para a entrega tardia do digital, para termos uma ideia da área, formato que queremos, pinos que queremos para conseguirmos avançar o trabalho da integração enquanto não tem o layout pronto.

A entrega do layout digital acontece muito tarde, muito perto do tapeout e sempre tem um probleminha, como DRC ou LVS, roteamento não obedeceu alguma distancia... Já aconteceu muito que o deadline do digital quase coincidia com o deadline do projeto, gerando um trabalho muito excessivo perto da entrega, virando noites... Gera um trabalho para os dois lados.

O DRC e LVS são rodadas no digital, mas não é o mesmo fluxo do analog. Rodar nas ferramentas do analog parece não estar mapeado no fluxo digital, o que acaba gerando problemas depois, geralmente bem na frente do projeto quando tempos pouco tempo. Poderia ser feito um trial do fluxo para já ir corrigindo esse problemas no início do projeto.

De LVS, cada memória nova trás dificuldades, a forma de fazer LVS é diferente. As vezes dão netlist pronto, as vezes esquemático, as vezes contem um dispositivo que não está no PDK... Em um projeto tentamos rodar LVS da memória separadamente para adiantar esses problemas. Também fazemos algo assim para IO cells. Essas coisas na maioria das vezes vem dentro do digital, mas como chegava tarde, acabamos adiantando essas atividades.

Um problema classico é as vias do digital em que temos que fazer attach da tecnologia do digital. Esse problema só é indentificado quando exportamos e importamos o GDS final (as vias somem), e temos que lembrar de fazer isso, deixar isso na cabeça.

Acabamos que temos duas ferramentas de versão em paralelo. O projeto não fica auto contido, a biblioteca do digital fica no SVN e um updated do SOS não atualiza o SVN. Para resolver, o gerente fazia um checkout local do SVN e ficava responsável por atualizar quando tinha entregas novas do digital. Isso resolveu essas mudanças.

Geralmente tem aquela reunião no começo (kick-off) que explica mais ou menos o que vai ser o projeto. Em projetos novos, é uma ideia abstrata. Quando vai desenvolvendo, entrando nos detalhes e as decisões vão sendo feitas, lá pro meio ou final do projeto, você não necessariamente ainda sabe

o que tá acontecendo, só o pessoal que tá trabalhando na integração ou simulação de topo. Tem projetos que mesmo assim quem tá no topo não sabe exatamente como funciona o digital, só o basico.

Essa perda de visão do topo ao longo do projeto é indiferente dependendo do caso. Seria legal saber, mas não vai ter um efeito negativo no geral do projeto. Pode haver casos e casos... se alguém for mudar algo no seu bloco, pode afetar outro no nível acima... como adicionar um bit de trimming quando o registrador separado para isso já não tem espaço...

Mais bits de trimming é normal, então geralmente superestimamos a quantidade de registradores. Em seguintes versões pode ser removido.

Saber suas restrições é mais importante do que ter um conhecimento do topo, estar livre para achar soluções tendo um limite de interface que pode usar e discutindo restrições. Essas restrições geralmente são estimadas no início das discussões baseadas em experiencias anteriores e mudanças são discutidas posteriormente.

A comunicação várias vezes é feita com o gerente como intermedio pois ele tem uma visão dos dois lados. Dificilmente as duas equipes se conversam, a não ser no nível mais de topo (simulação, layout) e mais ao final do projeto. Não sei se é porque não precisa ou se não é incentivado, porque dependendo do que estiver trabalhando não precisa. Um cara de bloco que precise de um bloquinho digital, por exemplo, até pode falar com alguém do time, mas geralmente comunica com o gerente.

Essa comunicação horizontal não está acontecendo. Não sei se precisa, mas acho que também há uma resistencia maior por exemplo por causa da linguagem diferente. Não há entendimento de como funciona o fluxo da outra equipe. Eu tenho uma certa noção, mas os detalhes e as dificuldades de certas demandas não. Se soubesse, poderia ter uma troca maior para pedir ajuda.

Informação do digital geralmente tá atualizada, mas não necessariamente sei onde procurar. Tenho dificuldade de navegar e encontrar uma informação específica. Tem vários diretórios, muitas coisas... não sei onde tá cada informação. É raro eu precisar, mas geralmente é algo relacionado ao topo como o endereço dos registradores. Acabou sendo mais facil olhar no Verilog e ficou mais fácil de ler.

As atividades de topo (integração, verificação), hoje em dia, é feita mais cedo. Já apanhamos muito com isso. Quando tem uma pessoa dedicada ao topo (e não com responsabilidade dividida) as atividades são feitas antes e essas atividades não ficam para depois ou em segundo plano. É bom para apertar as equipes para definir algumas coisas mais cedo. Coisas definidas superficialmente no início são refinadas para já ir pensando nas consequencias. Isso me parece funcionar melhor nos ultimos projetos que participei.

17 - Analog 06

Interview – 30/09/21

< Only annotation of points that are not already very saturated >

Eu entrei quando o projeto já estava rodando, tinha muita coisa que eu não entendia. Fui entendendo aos poucos o projeto. Eu sabia o que era mais crítico, o que tinha que tomar de conta, a parte de interface não mudava muito. A falta de visão de topo não atrapalhou nesse projeto.

Eu acho muito importante ter essa visão de topo, mas em reuniões só de topo fica centralizada em poucas pessoas para não tomar tempo das pessoas não diretamente envolvidas. Mas seria bom outros participarem para desenvolver essa habilidade e preparar para os próximos projetos.