[13:03] ccberni (Convidado)

Rodrigo, boa tarde, tudo bem?

​

[13:04] ccberni (Convidado)

Estou iniciando o projeto da alteração do codec spw\_light, gostaria de ver alguns detalhes contigo...

​

[13:04] ccberni (Convidado)

Vou fazer um resumo aqui, aí você fica à vontade para comentar, ok?

​

[13:09] ccberni (Convidado)

1. Vocês estão usando a versão que está no opencores, sem alterações? Se tiver alteração, me envie o pacote, por favor...
2. A interface usada continua sendo a stream apenas, certo?
3. Pretendo trabalhar ampliando os sinais de interface do ip, para que seu módulo de controle possa comandar a emissão dos erros; isso não seria feito via "wrapper" externo, e sim diretamente no ip do spw\_light. Tudo bem pra vocês ter uma versão customizada?
4. Devo trabalhar diretamente no modelsim, com os testbenchs que já existem, e talvez com outros mais. No caso do link spw, a ideia é fechar um loopback tx-rx.
5. Eu iria operar com uma pasta local aqui em casa, com git local, e depois "subir" num branch do NSEE. Se achar melhor fazer como no passado, e já criar um branch, ok também!
6. Por favor envie informações que achar importante, para estarmos sincronizados.

​

[14:04] Rodrigo de Marca França

Boa tarde! Tubo bem sim, e com você?

​

[14:04] Rodrigo de Marca França

1. Sim, sem alterações. Posso te enviar os arquivos do projeto, mas é o codec original;
2. Sim, usamos a interface spwstream;
3. Acredito que não tenha nenhum problema. Modificar a interface dele para suportar isso me parece bem razoável. Precisamos depois ver como seria possível “válidar” o codec modificado. Talvez seja possível pegar o Conformance Tester emprestado com o CTI;
4. Sem problemas. Quando for o momento de sintetizar um HW a gente vê como podemos fazer a integração no sistema;
5. O que você achar melhor. Se quiser já deixar um branch remoto, ou eventualmente até um repositório novo só para isso, fique à vontade. Mas não vejo nenhum problema você trabalhar localmente e colocar depois em um branch. Se puder, dá uma perguntada para o Érico o que ele acha melhor, mas do meu ponto de vista não teria nenhum problema;
6. Eu não sei exatamente o que ficou combinado com você, mas entendo que a lista de erros está bem definida no URD (se ficou alguma dúvida, pode perguntar). Vou listar algumas observações gerais que eu tenho de cabeça, mas quero deixar claro que não são diretivas de implementação. Deixo a seu critério definir a melhor maneira de fazer a interface e o funcionamento:
   1. Eu pensei nos erros serem “trigados”, ou seja, o usuário dar um pulso na entrada e esse pulso causar apenas um erro daquele tipo (não ser algo como liga e desliga o erro);
   2. Acredito que erros trigados em um momento “errado” ou que não faça sentido possam ser desconsiderados (por exemplo: erros no link enquanto o link já está desconectado);
   3. Não acho necessário gerenciar múltiplos erros simultaneamente. Se o usuário mandar vários erros serem trigados ao mesmo tempo, pode cair em uma estrutura de if-elsif-else e executar apenas um deles;
   4. Pode ter algum flag de “error state” ou “injecting error” para indicar para o usuário que tem um erro sendo injetado/esperando ser injetado. Não seria necessário, mas pode ser útil para evitar que o usuário tente trigar um erro enquanto outro está acontecendo, ou em um momento incorreto.

Editada

​

[14:50] ccberni (Convidado)

Show, Rodrigo, valeu pelas respostas e informações, manterei contato por aqui. Grande abraço!