



RELATÓRIO DE ESTÁGIO

Migração de uma cadeia de comunicação integrando o protocolo HARQ para um microssistema embarcado que dialoga através de um NoC

ALUNO: **Daniel Victor Silva de Oliveira**

ORIENTADORES: **Christine Hennebert – CEA**
Prof. Hiran de Melo – UFCG
Prof. Dominique Tournier – INSA

LOCAL: **CEA / Leti – Grenoble, França**

PERÍODO: **Abril a Setembro de 2008**



*Migração de uma cadeia de comunicação integrando o
protocolo HARQ para um microssistema embarcado
que dialoga através de um NoC*

**Estágio de abril a setembro de 2008 no
CEA / Leti – Grenoble, França.**

Daniel Victor Silva de Oliveira

Prof. Hiran de Melo

Campina Grande, novembro de 2009

Agradecimentos

A DEUS, em primeiro lugar, por sua Luz e proteção divinas.

A Christine Hennebert, por ter me acolhido nesse estágio e por ter me conduzido e aconselhado com seriedade e motivação.

A Hiran de Melo, por ter me orientado com maestria neste trabalho.

A Dominique Tournier, por ter me dirigido e acompanhado ao longo do estágio.

A Andrezza Dantas, minha namorada, por seu apoio moral e por suas palavras tenras de encorajamento.

A minha família, sempre presente nos meus projetos, por seu apoio incondicional.

Índice

Lista de Abreviaturas.....	5
Lista de Figuras.....	6
Lista de Tabelas.....	7
1 Introdução.....	8
2 Apresentação do CEA.....	10
CEA de Grenoble.....	10
Leti.....	11
DCIS.....	11
SASTI.....	12
LASP.....	12
3 Plano do Estágio.....	13
4 Protocolo HARQ.....	14
5 Plataforma Física.....	16
6 Aplicação Estudada.....	19
7 Sistema Existente.....	22
8 Novo Sistema.....	28
9 Análise dos Resultados.....	35
10 Conclusão.....	39
Bibliografia.....	40
Anexo: Caminhos de Roteamento.....	41

Lista de Abreviaturas

ACK	Acknowledgement
AHB	Advanced High-speed Bus
ARQ	Automatic Repeat reQuest
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BoP	Begin of Packet
CBR	Constant Bit Rate
CEA	Commissariat à l’Energie Atomique
CRC	Cyclic Redundancy Check
DCIS	Département Conception et Intégration Système
DRT	Direction de la Recherche Technologique
EoP	End of Packet
FAUST	Flexible Architecture of Unified System for Telecom
FEC	Forward Error Correction
FPGA	Field Programmable Gate Array
HARQ	Hybrid Automatic Repeat reQuest
IHM	Interface Homem-Máquina
IP	Intellectual Property
IP	Internet Protocol
IT	Interrupção
LASP	Laboratoire Architectures numériques, Sécurité des composants et Plates-formes de validation
Leti	Laboratoire d’électronique et de technologie de l’information
NACK	Negative Acknowledgement
NoC	Network-on-Chip
PC	Personal Computer
RAM	Random Access Memory
Rx	Recepção
SASTI	Service Architecture et Sécurité pour les Technologies de l’Information
Tx	Transmissão
UMTS	Universal Mobile Telecommunications System
UWB	Ultra-Wideband
VHDL	Very high speed integrated circuit Hardware Description Language

Lista de Figuras

Figura 1: Stop and Wait ARQ.....	15
Figura 2: Arquitetura interna da plataforma física.....	16
Figura 3: Fotografia da plataforma	17
Figura 4: O processador ARM e seus periféricos	17
Figura 5: Topologia do NoC.....	18
Figura 6: Cadeia de comunicação	20
Figura 7: Sistema e ferramentas de desenvolvimento.....	21
Figura 8: Organograma simplificado do programa do PC	23
Figura 9: Organograma simplificado do programa do ARM	24
Figura 10: Interface homem-máquina do sistema existente	25
Figura 11: Diagrama de fluxo – inicialização.....	25
Figura 12: Diagrama de fluxo – “roteiro ACK”	26
Figura 13: Diagrama de fluxo – “roteiro NACK”	26
Figura 14: Curva de taxa de erro.....	27
Figura 15: Disposição das memórias associadas ao processador ARM.....	30
Figura 16: Organograma simplificado do novo programa do ARM.....	31
Figura 17: Organograma simplificado do novo programa do PC.....	32
Figura 18: Novo diagrama de fluxo – inicialização	32
Figura 19: Novo diagrama de fluxo – “roteiro ACK”.....	33
Figura 20: Novo diagrama de fluxo – “roteiro NACK”.....	33
Figura 21: Interface homem-máquina do novo sistema	34
Figura 22: Nova curva de taxa de erro	35
Figura 23: Tempo de transmissão do sistema existente	36
Figura 24: Tempo de transmissão do novo sistema	37
Figura 25: Tempo de transmissão dos dois sistemas	37

Lista de Tabelas

Tabela 1: Cronograma do estágio	13
Tabela 2: Códigos das direções para o caminho de roteamento	18
Tabela 3: Distribuição de tarefas entre o PC e o processador ARM.....	22
Tabela 4: Nova distribuição de tarefas entre o PC e o processador ARM	28
Tabela 5: Organização da memória externa	29
Tabela 6: Organização da memória RAM 2	29
Tabela 7: Arquivo de referência para as medições de tempo de transmissão	35
Tabela 8: Tempo de transmissão do sistema existente.....	36
Tabela 9: Tempo de transmissão do novo sistema.....	36
Tabela 10: Caminhos de roteamento.....	41

1 Introdução

Hoje em dia, as normas de telecomunicações avançam rapidamente, donde a pertinência e o interesse de se desenvolver uma plataforma física capaz de suportar vários protocolos de comunicação e de integrar algoritmos complexos, incluindo reconfigurações dinâmicas.

Nesse contexto de evolução permanente, a reutilização de unidades funcionais (IPs, sigla em inglês) genéricas permite encurtar o tempo de concepção e de lançamento no mercado de novos produtos.

As exigências em termos de taxa de transmissão estão cada vez mais elevadas e conduzem a otimizar os tempos de transferência e de tratamento no interior dos sistemas. Os NoCs (Rede-em-Chip, sigla em inglês) aparecem como uma alternativa promissora, pois eles revolucionaram o modo de comunicação habitual por barramento. Eles permitem, na verdade, trocas de informação mais rápidas e sobretudo em paralelo.

Atualmente, o barramento partilhado é o meio de comunicação mais usual. Trata-se de um único canal físico conectado aos diferentes componentes do sistema, que se comunicam via um protocolo definido. Ele é flexível, reutilizável e a adição de um novo recurso é simples. Em contrapartida, a partilha do barramento engendra tempos de espera, pois ele não está sempre disponível para os IPs. Além disso, em função do número de componentes, o caminho para acessar um bloco determinado pode ser longo, o que implica um atraso de transmissão maior e um grande consumo de energia. Essa técnica é mais adaptada aos sistemas compostos de um número limitado de unidades.

Os barramentos hierárquicos, outra solução de comunicação, ligam vários barramentos entre si através de pontes. Cada um agrupa recursos que se comunicam mais frequentemente e possui seu próprio árbitro. Todavia, a circulação de informação entre barramentos diferentes faz crescer consideravelmente a complexidade do sistema.

Na comunicação ponto a ponto, as ligações são específicas e dimensionadas para cada conexão, e o controle de fluxo é ainda mais fácil de implementar.

No caso dos NoCs, a conexão do bloco à rede é uniforme e independente de sua atribuição, o que favorece a adição de novos elementos. Finalmente, eles representam o melhor compromisso tempo de desenvolvimento/área necessária/consumo de energia. As Redes-em-Chip caracterizam arquiteturas flexíveis, extensíveis e reconfiguráveis, e satisfazem bem às exigências das comunicações sem fio multipadrão.

O objetivo principal do estágio é a migração de uma cadeia de comunicação integrando o protocolo HARQ (Hybrid Automatic Repeat reQuest) para um microsistema embarcado que dialoga através de um NoC, visando o desenvolvimento de aplicações em “tempo real”.

No presente relatório, é apresentado inicialmente o local onde se realizou o estágio e o plano de trabalho. Em seguida, é realizada uma breve explanação sobre o protocolo HARQ, sobre a plataforma física utilizada e sobre a aplicação estudada. Logo depois, o sistema existente e o novo sistema desenvolvido são descritos em detalhes. Por fim, os resultados obtidos são analisados e algumas considerações finais são tecidas, além de sugestões para possíveis trabalhos futuros.

2 Apresentação do CEA

O CEA (Comissariado de Energia Nuclear, sigla em francês) foi fundado em 18 de outubro de 1945 por ordem do general De Gaulle. No dia seguinte da segunda guerra mundial, esse organismo deveria permitir à França ganhar sua independência na área nuclear. O papel do CEA é então assegurar as pesquisas na área nuclear para aplicações industriais, médicas e militares.

Hoje, o CEA é, com seus 15.612 assalariados distribuídos em nove centros e com um orçamento de 3,4 bilhões de euros, um ator maior em matéria de pesquisa, de desenvolvimento e de inovação. Ele intervém em três grandes áreas: a energia, as tecnologias para a informação e para a saúde, a defesa e a segurança. O CEA se organiza em torno de quatro polos: programas para a defesa, pesquisa fundamental, pesquisa e tecnologia para a indústria e P & D para a energia nuclear.

CEA de Grenoble

O CEA de Grenoble foi criado em 1956 pelo Prêmio Nobel de Física Louis Néel. Ele é, com seus 2.300 assalariados e com um orçamento de 350 milhões de euros, o primeiro centro de pesquisa na Região Rhône-Alpes e o principal do polo pesquisa tecnológica do CEA.

No seio do CEA de Grenoble, a Direção da Pesquisa Tecnológica (DRT, sigla em francês) tem por missão fundamental identificar e desenvolver, em colaboração com os outros polos e com os industriais, as novas tecnologias da informação e da comunicação nas áreas das micro e nanotecnologias e dos sistemas a software predominantemente.

Leti

O Leti (Laboratório de eletrônica e de tecnologia da informação, sigla em francês) foi criado em 1967 e faz parte da DRT, possuindo hoje um efetivo de 1.000 assalariados CEA e 500 colaboradores externos. Seu orçamento é de aproximadamente 174 milhões de euros por ano, ou seja, mais de um terço daquele do CEA de Grenoble.

As missões do Leti são: ajudar as empresas a aumentar sua competitividade graças à inovação tecnológica e assegurar a transferência de seu conhecimento para a indústria. Ele consagra também 85% de sua atividade a pesquisas finalizadas com parceiros externos ao CEA. Suas principais áreas de atividade são: a microeletrônica; tecnologias, concepção e integração dos microssistemas; tecnologias da imagem; micro e nanotecnologias para a biologia e para a saúde; tecnologias de comunicação e objetos nômades.

DCIS

No seio do Leti, o Departamento Concepção e Integração Sistema (DCIS, sigla em francês) é encarregado das pesquisas, dos estudos e do desenvolvimento das aplicações industriais nas áreas dos sistemas e microssistemas e dos sensores.

O DCIS contribui à orientação dos novos desenvolvimentos tecnológicos da DRT e participa na atividade de difusão tecnológica junto a pequenas e médias empresas e a pequenas e médias indústrias no campo da eletrônica.

A organização do DCIS é baseada no desenvolvimento de projetos, cada um sendo frequentemente ligado a um tema de pesquisa.

SASTI

O Serviço Arquitetura e Segurança para as Tecnologias da Informação (SASTI, sigla em francês) faz parte do DCIS e desenvolve suas atividades de pesquisa e de desenvolvimento nas áreas do tratamento do sinal e dos algoritmos, das arquiteturas digitais, da segurança dos componentes e das plataformas de validação e da avaliação da segurança das tecnologias da informação. O serviço é dividido em três laboratórios, entre eles o LASP.

LASP

O Laboratório Arquiteturas digitais, Segurança dos componentes e Plataformas de validação (LASP, sigla em francês) conduz em sua maior parte trabalhos sobre arquiteturas eletrônicas digitais de tratamento da informação. O laboratório atua nas áreas do tratamento digital de bandas de base, no desenvolvimento das técnicas de transmissão por fracionamento de espectro (UWB), assim como nas técnicas de transmissão para o grande público (UMTS, Wi-Fi). Com a ajuda de plataformas de validação, o laboratório avalia os desempenhos dos sistemas que ele próprio realiza.

3 Plano do Estágio

O objetivo principal do estágio é a migração de uma cadeia de comunicação integrando o método HARQ para um microssistema embarcado que dialoga através de um NoC. A intenção é tornar o sistema existente totalmente independente e melhorar as taxas de transmissão sem deteriorar o seu desempenho. Futuramente, a execução em tempo real da aplicação será desejada.

O estágio pode ser dividido em quatro grandes fases:

- Fase de estudo – ela compreende o estudo bibliográfico, o aprendizado das ferramentas de desenvolvimento e, posteriormente, a medição e a análise dos resultados obtidos;
- Fase de preparação – ela engloba a alocação das memórias do sistema, a formatação dos arquivos de configuração e a elaboração dos códigos para o processador ARM;
- Fase de implementação – ela inclui o desenvolvimento da nova interface homem-máquina (IHM), a implementação da aplicação e sua depuração;
- Fase de redação dos relatórios e preparação da apresentação.

O cronograma efetivamente seguido é apresentado na Tabela 1.

Tabela 1: Cronograma do estágio

TAREFAS	Semanas																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Fase de estudo																							
Estudo bibliográfico	█									█													
Aprendizado das ferramentas de desenvolvimento		█	█	█	█					█	█												
Medição e análise dos resultados																					█	█	
Fase de preparação																							
Alocação das memórias						█											█						
Formatação dos arquivos de configuração								█	█								█						
Elaboração dos códigos ARM								█	█					█					█				
Fase de implementação																							
Desenvolvimento da IHM														█	█	█	█	█		█	█		
Implementação e depuração da aplicação														█	█	█	█			█			
Redação dos relatórios					█	█			█													█	█

4 Protocolo HARQ

Existem dois tipos de mecanismos de controle de erros em uma transmissão:

- Reativos (ARQ) – o receptor deve sinalizar se o pacote transmitido está íntegro e, assim, o emissor responderá adequadamente;
- Proativos (FEC) – no momento da emissão, informações de redundância são adicionadas ao pacote a ser transmitido, de forma a permitir que o receptor corrija eventuais imprecisões.

A associação de ambos dá origem ao protocolo HARQ, que conserva os benefícios de cada um.

Existem três modos de realizar o protocolo ARQ:

- Stop and Wait – o emissor envia um pacote de dados e aguarda uma resposta sobre a integridade dele no momento da recepção. Se ele não estiver exato ou se nenhuma resposta é recebida, um reenvio é realizado. Um temporizador é utilizado para determinar um tempo máximo de espera.
- Go Back N – vários pacotes sucessivos são transmitidos sem aguardar uma resposta do receptor. Quando um pacote não é quitado, ele é reenviado, assim como todos os pacotes seguintes já transmitidos. A ordem dos pacotes é conservada. Um buffer é necessário no lado do emissor.
- Selective Repeat – segue o mesmo princípio do método precedente, a diferença é que apenas o pacote incorreto é retransmitido. Portanto, a ordem dos pacotes não é mais conservada. Por conseguinte, a presença de um buffer no lado do receptor é igualmente indispensável.

O principal inconveniente do método Stop and Wait (Figura 1) é sua dependência ao tempo de espera da sinalização. No entanto, ele é o mais simples, conserva a ordem dos pacotes transmitidos, não precisa de um grande espaço de memória, e apenas os pacotes incorretos são retransmitidos. Além do mais, para atenuar o inconveniente de tempo, é possível instanciar N processos Stop and Wait independentes e simultâneos.

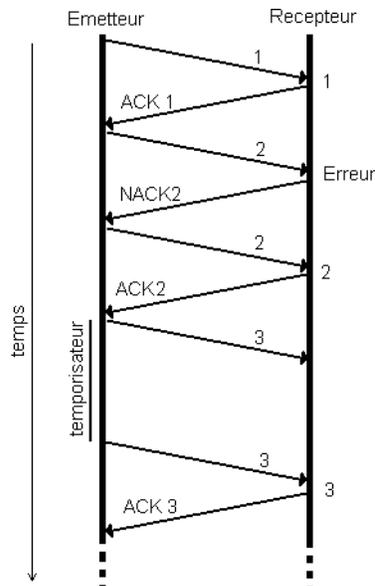


Figura 1: Stop and Wait ARQ

O protocolo HARQ compreende a detecção de pacotes incorretos e a combinação de várias retransmissões a fim de restabelecer a integridade deles. Esta é verificada graças aos bits CRC (Verificação de Redundância Cíclica, sigla em inglês) adicionados no final dos dados transmitidos. Esses bits são calculados, no momento da emissão, a partir do pacote de dados e de um polinômio gerador. A decodificação deles na recepção permite avaliar a integridade dos dados e decidir se uma retransmissão será solicitada. A adição de um bloco FEC, capaz de corrigir alguns erros, permite reduzir o número de retransmissões.

Diversas técnicas de combinação são possíveis:

- Chase Combining – a cada retransmissão, o pacote é transmitido com os mesmos parâmetros de codificação.
- Incremental Redundancy – a cada retransmissão, blocos diferentes, não necessariamente de mesmo tamanho, são transmitidos. Eles não são decodificáveis sem o pacote da primeira transmissão, composta de bits sistemáticos.
- Incremental Redundancy parcial – é uma combinação dos dois casos precedentes; os pacotes retransmitidos têm o mesmo tamanho e são decodificáveis independentemente dos outros, assim como novos bits de paridade são emitidos a cada retransmissão.

O sistema implementado é baseado no protocolo HARQ utilizando a técnica chase combining.

5 Plataforma Física

A aplicação estudada é desenvolvida sobre uma plataforma física (Figura 3) composta de dois ASICs (Circuito Integrado de Aplicação Específica, sigla em inglês) chamados FAUST (à direita na Figura 2) e de dois FPGAs (Arranjo de Portas Programável em Campo, sigla em inglês) Xilinx Virtex 4 (à esquerda na Figura 2). Possuindo recursos dedicados ao tratamento e à transmissão de dados, o circuito FAUST foi concebido com a tecnologia NoC para aplicações de telecomunicações. Os componentes reprogramáveis estão conectados a uma extensão da rede embarcada nos ASICs, o que permite integrar novos IPs, que se comunicarão portanto via o mesmo protocolo.

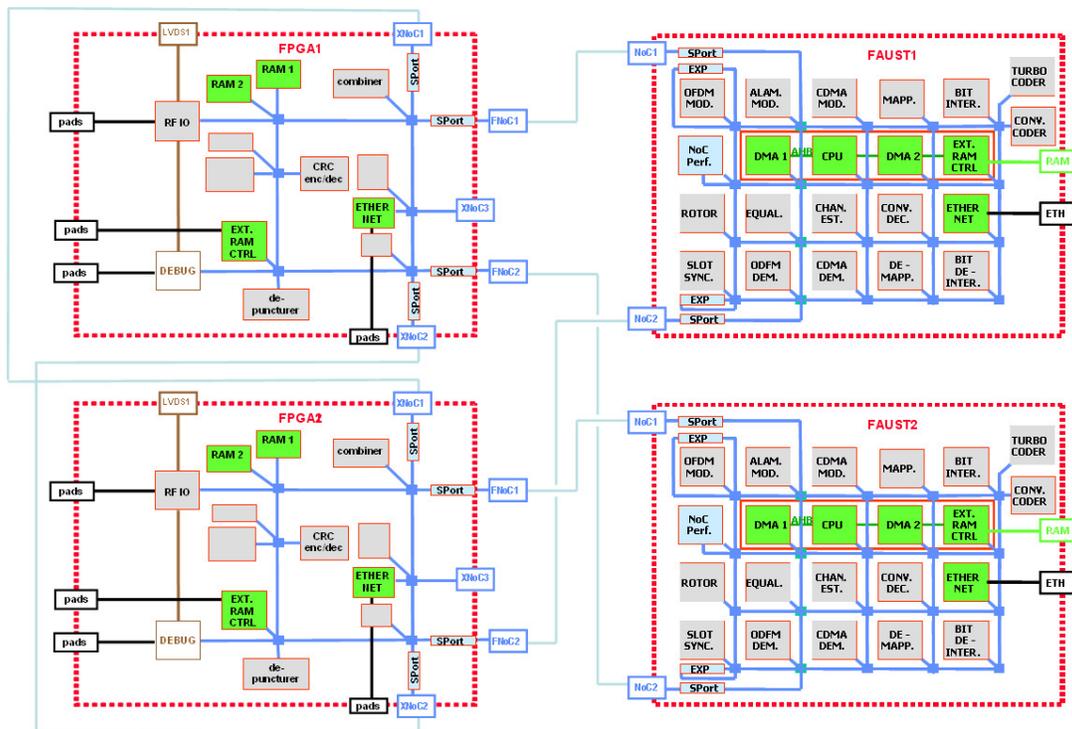


Figura 2: Arquitetura interna da plataforma física



Figura 3: Fotografia da plataforma

O controle do circuito FAUST e de suas extensões é efetuado por um processador ARM946E-S (Figura 4), que dispõe de duas memórias RAM (Memória de Acesso Aleatório, sigla em inglês) internas de 128Kb e de uma memória externa de 8Mb, acessível através de um controlador de memória. O processador é ligado aos seus periféricos por um barramento AHB (Advanced High-speed Bus) e à Rede-em-Chip por uma interface que realiza a conversão de protocolo “AHB \Leftrightarrow NoC”. A comunicação com o exterior é possível graças a uma porta Ethernet; uma unidade específica é encarregada da conversão de protocolo “Ethernet \Leftrightarrow NoC”.

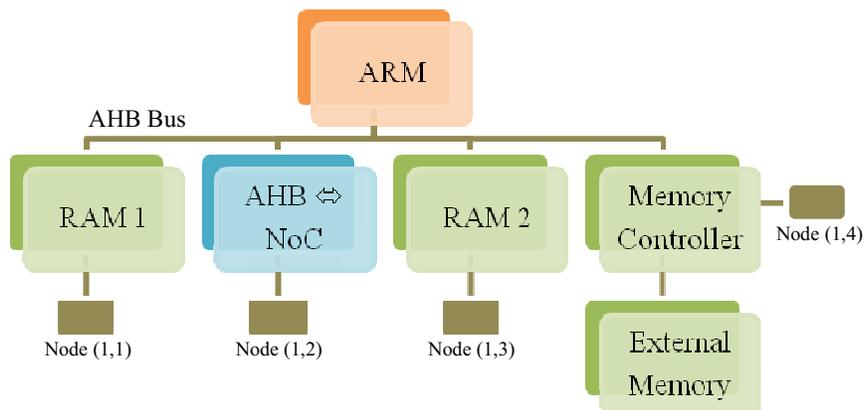


Figura 4: O processador ARM e seus periféricos

A Rede-em-Chip é uma malha 2D de 20 nós (Figura 5), cada um possuindo 5 portas bidirecionais: 4 para a comunicação com os nós vizinhos (norte - N, sul - S, leste - E e oeste - W) e 1 para a conexão com um recurso. Ela utiliza um roteamento determinista do tipo fonte: o caminho a percorrer é definido pelo emissor e permanece fixo em toda a transmissão, desde a fonte até o destino.

Os dados são divididos em pacotes, cada um contendo várias palavras (*flits*) de 32 bits. O primeiro *flit* do conjunto é o cabeçalho (*header*), informando a natureza das informações e o seu roteamento. Um campo de 18 bits é reservado para codificar o caminho. Cada direção é codificada em 2 bits (Tabela 2), assim é possível encadear nove direções sucessivas. É proibido dar “meia-volta”. Por convenção, quando voltamos (retomamos a direção de proveniência), entramos no recurso conectado ao nó. Os caminhos são pré-calculados pelo projetista e carregados nos IPs, a fim de evitar bloqueios. Dois bits adicionais, indicando o início e o fim do pacote, são colocados antes de cada palavra: o BoP (Begin of Packet) e o EoP (End of Packet). Os *flits* de dados trafegando na rede totalizam, portanto, 34 bits ao invés de 32.

Tabela 2: Códigos das direções para o caminho de roteamento

Direção	N	E	S	W	Recurso
Código	00	01	10	11	<i>direção proibida*</i>

**ex.: 01 se o pacote provém do leste*

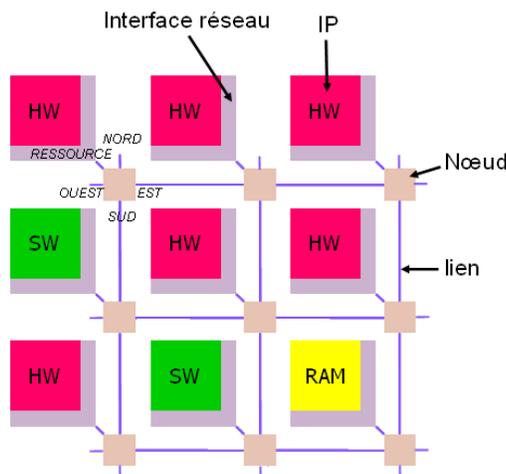


Figura 5: Topologia do NoC

6 Aplicação Estudada

Uma cadeia de comunicação completa (Figura 6) baseada no protocolo HARQ “Stop and Wait” foi implementada na plataforma física. Ela é composta de buffers, de unidades encarregadas da detecção de erros (codificador e decodificador CRC), de blocos responsáveis pela codificação e pela decodificação da informação (codificador convolucional, entrelaçador, mapeador e seus respectivos correspondentes), de um módulo que emula o canal de transmissão AWGN (Ruído Aditivo Gaussiano Branco, sigla em inglês) e de um recurso, chamado combinador HARQ, que combina os pacotes retransmitidos quando um erro é identificado, a fim de restabelecer a integridade deles.

Ao fim de uma transmissão, o decodificador CRC sinaliza o estado dos dados recebidos. Se eles estão isentos de erro, um sinal de reconhecimento positivo (ACK, sigla em inglês) solicita a emissão de um novo pacote. Quando o sinal de reconhecimento é negativo (NACK, sigla em inglês), os dados são memorizados para a combinação e o reenvio do mesmo pacote é solicitado. É importante observar que, nesta aplicação, o sinal ACK/NACK não sofre, hipoteticamente, os efeitos do ruído do canal.

A maioria das unidades funcionais da aplicação já está presente nos ASICs. Outras foram desenvolvidas e integradas nos componentes programáveis, com o auxílio das ferramentas Xilinx.

A concepção arquitetônica da plataforma prevê a utilização do circuito FAUST1 (f1) e do FPGA1 (x1) para realizar a transmissão (Tx), enquanto que o circuito FAUST2 (f2) e o FPGA2 (x2) se ocupam da recepção (Rx).

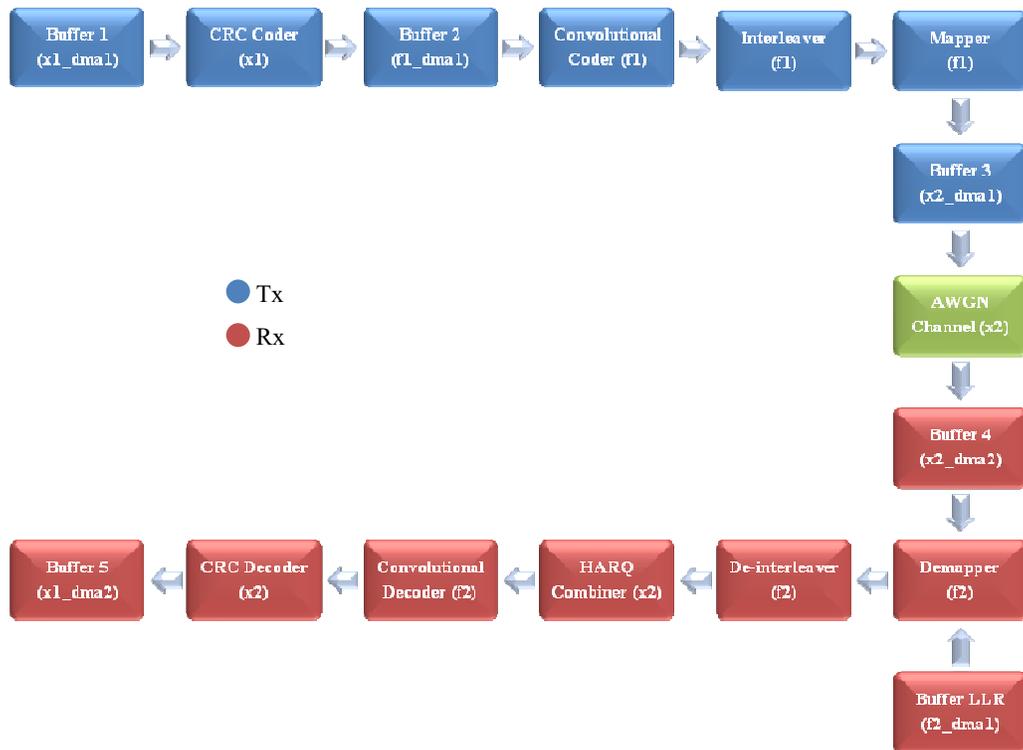


Figura 6: Cadeia de comunicação

O controle do sistema compreende a inicialização da aplicação e a gestão do protocolo HARQ: a configuração dos IPs e o tratamento das interrupções (ITs). Ele é dividido entre um dos processadores ARM, o de f2, e um computador pessoal (PC, sigla em inglês). Este é ligado à x1 pela interface Ethernet e um mecanismo de soquetes (Windows) permite as transferências. Ele pode acessar todos os recursos e receber ITs e fluxos de dados vindos do sistema. A comunicação é regida por uma interface homem-máquina desenvolvida em Visual C++. Uma linha de execução (*thread*) permite que o PC esteja sempre em escuta, para poder aceitar os dados e reagir a todas as ITs chegando a qualquer momento.

Antes de implementar a aplicação na plataforma física, as etapas de co-simulação (SystemC + VHDL) do sistema e de emulação (Assembly) do ARM são necessárias. A primeira, realizada em Modelsim, é importante para ajustar a aplicação e gera os arquivos de configuração das memórias e dos blocos funcionais. A outra, efetuada com o programa ARMulator, permite prever o comportamento do processador quando da execução e fornece o seu código de inicialização e o programa gestor da aplicação.

Em um sistema tão complexo (Figura 7), as ferramentas de depuração são fundamentais. Atualmente, os sinais internos dos ASICs não são acessíveis; no entanto, através do programa ChipScope de Xilinx, é possível visualizar os sinais internos dos FPGAs. Para verificar as trocas entre o PC e a plataforma, a ferramenta CommView é capaz de mostrar todos os dados que transitam entre eles e os endereços IP (Protocolo de Internet, sigla em inglês) envolvidos. Além disso, pontos de parada podem ser colocados no código da IHM em Visual C++, o que permite pausar a aplicação, observar e eventualmente modificar o estado do sistema durante a execução. Finalmente, o ARMulator permite rodar e depurar o código destinado ao processador em um ambiente à parte e graças a um *test-bench* reproduzindo os pedidos e os comportamentos do sistema.

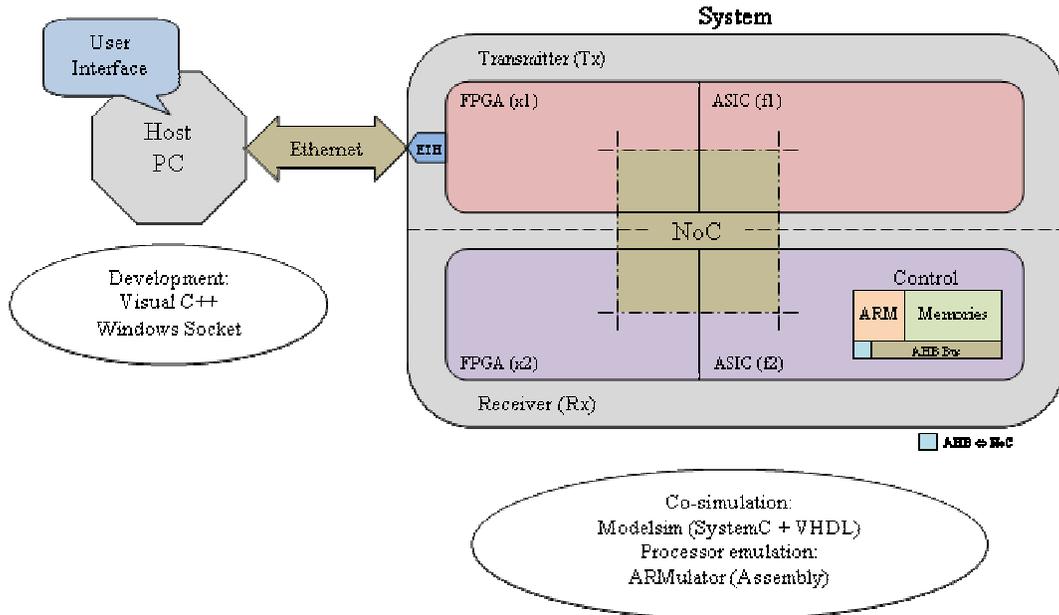


Figura 7: Sistema e ferramentas de desenvolvimento

7 Sistema Existente

Em uma primeira implementação do sistema, o PC desempenha o papel de mestre e o ARM, de escravo (Tabela 3). Uma vez lançada a aplicação, o PC é encarregado de enviar as configurações iniciais assim como os *flits* de reconfiguração aos IPs, de gerenciar as retransmissões, de fornecer os dados na entrada do sistema e de ler os pacotes transmitidos na saída. O processador se ocupa unicamente de coletar a IT ACK/NACK e de redirecioná-la para o mestre. Os organogramas simplificados dos programas dos dois atores são expostos nas Figuras 8 e 9.

Tabela 3: Distribuição de tarefas entre o PC e o processador ARM

	PC (mestre)	ARM (escravo)
T A R E F A S	<i>Definir o nível de ruído</i>	<i>Autoconfigurar-se</i>
	<i>Fixar o número máximo de retransmissões</i>	<i>Redirecionar a IT ACK/NACK para o mestre</i>
	<i>Carregar as memórias</i>	
	<i>Iniciar o processador e lançar a aplicação</i>	
	<i>Configurar a cadeia de comunicação</i>	
	<i>Fornecer os dados na entrada</i>	
	<i>Gerenciar as retransmissões</i>	
	<i>Recuperar os pacotes na saída</i>	

O conjunto de configurações que preparam a cadeia de comunicação para transmitir um novo pacote de dados é chamado “roteiro ACK”. De maneira análoga, o “roteiro NACK” configura a cadeia para retransmitir um mesmo pacote.

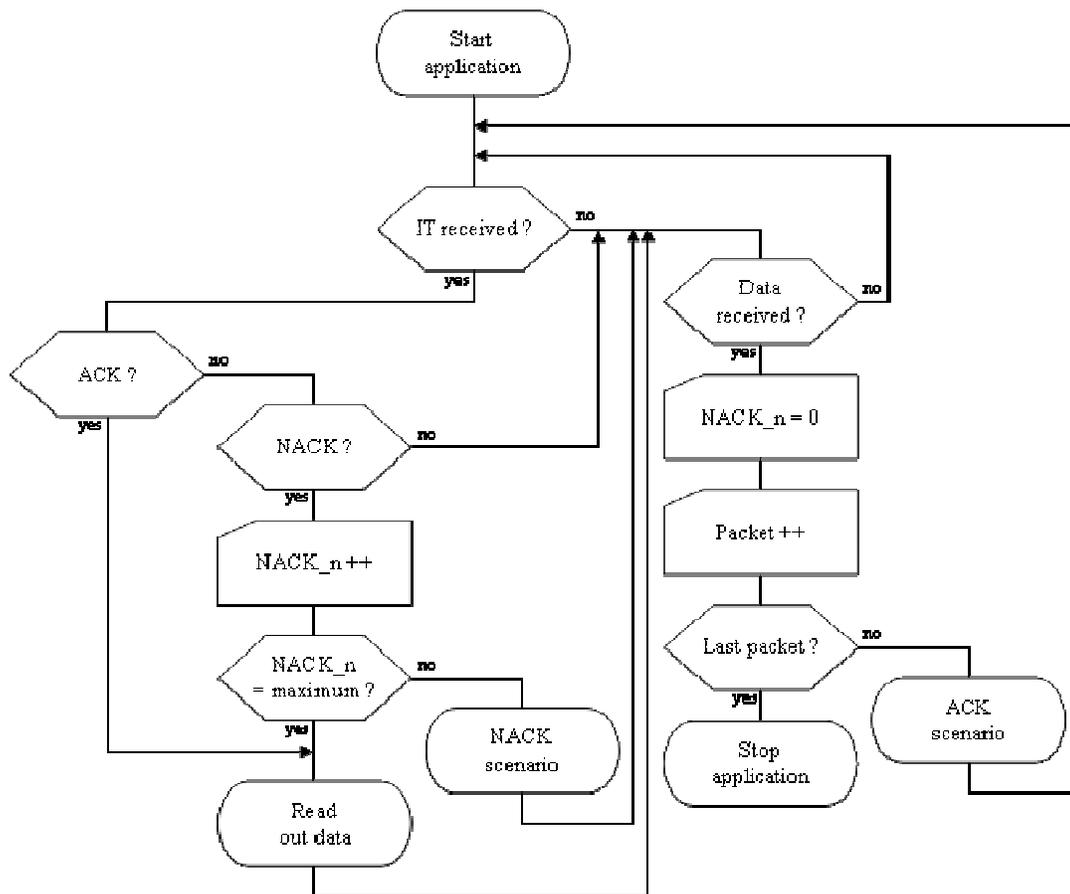


Figura 8: Organograma simplificado do programa do PC

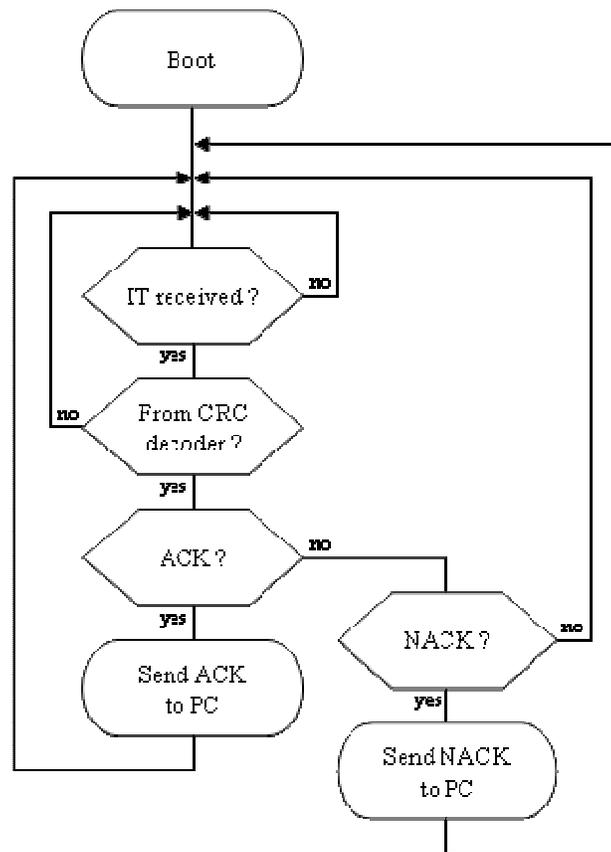


Figura 9: Organograma simplificado do programa do ARM

A IHM utilizada e os diagramas de fluxo descrevendo o funcionamento da aplicação, no momento da inicialização, do advento de um ACK e do advento de diversos NACK, são mostrados nas Figuras 10, 11, 12 e 13.

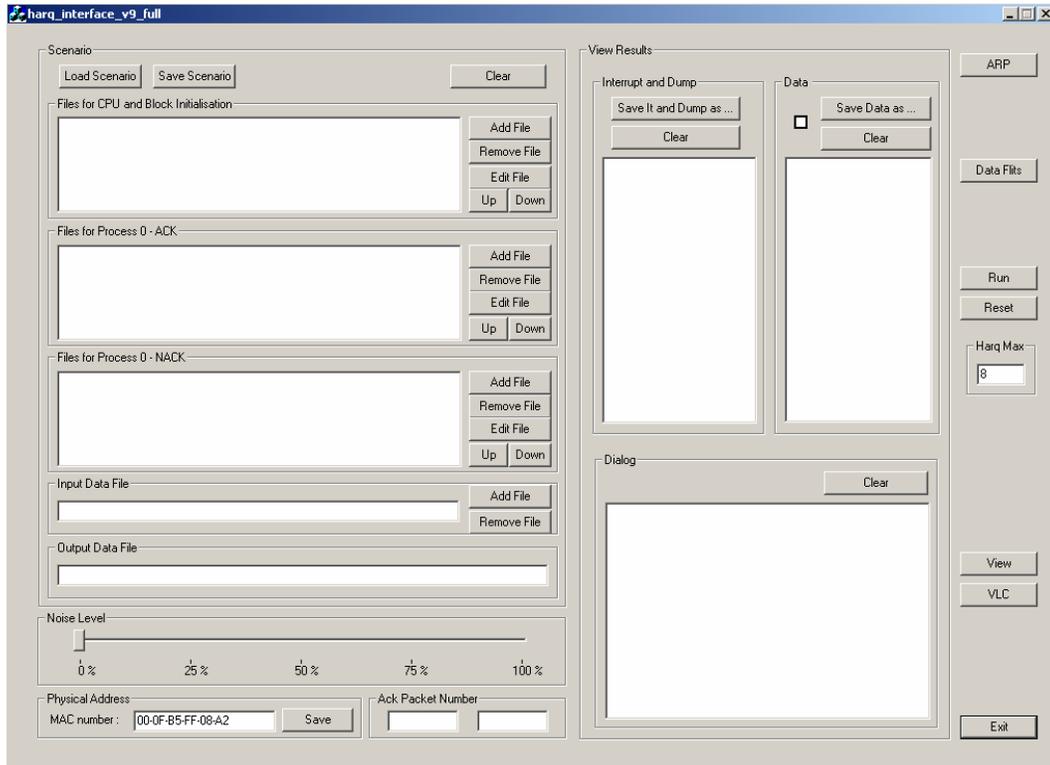


Figura 10: Interface homem-máquina do sistema existente

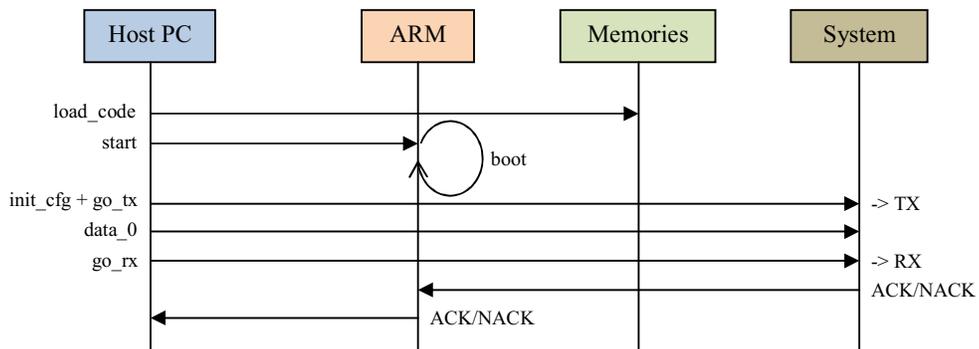


Figura 11: Diagrama de fluxo – inicialização

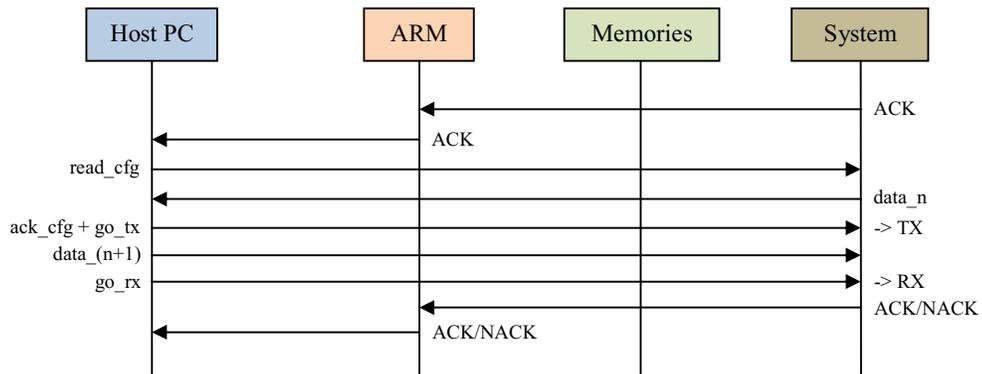


Figura 12: Diagrama de fluxo – “roteiro ACK”

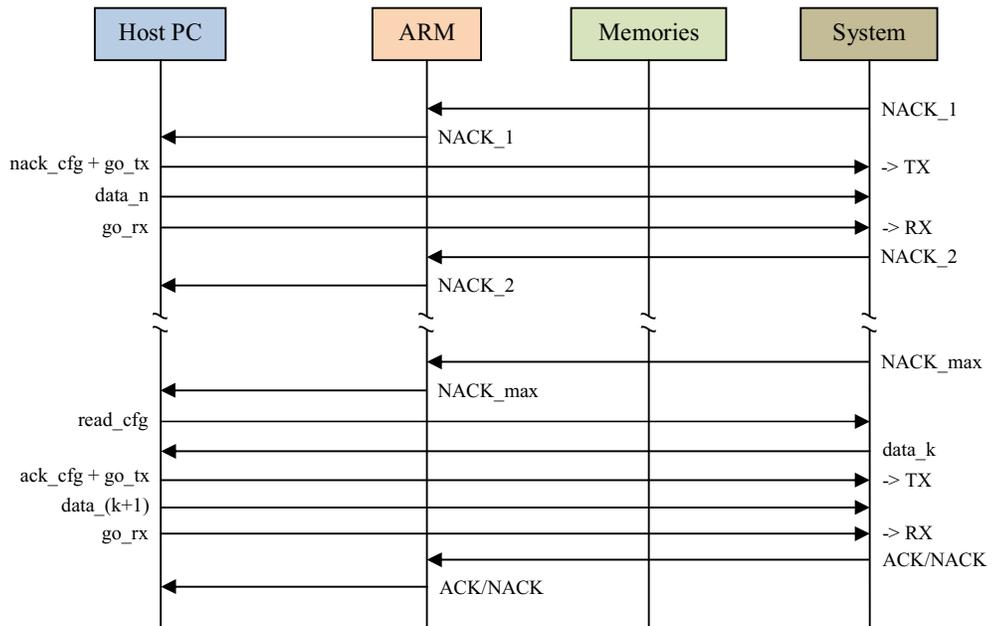


Figura 13: Diagrama de fluxo – “roteiro NACK”

O sistema, governado pelo PC, constitui uma etapa importante na metodologia de desenvolvimento. Ela permite fazer o ajuste fino de todas as configurações e de todas as trocas necessárias ao bom funcionamento da aplicação, para atingir os desempenhos desejados em termos de taxa de erro (BER, sigla em inglês). A curva BER, permitindo avaliar a qualidade da transmissão, é mostrada na Figura 14. A passagem da simulação realizada com números flutuantes a um sistema real funcionando com números quantificados provoca uma perda de 1dB. O fato de autorizar uma retransmissão pelo mecanismo HARQ permite ganhar 3dB. Esse resultado é muito interessante. O objetivo é explorá-lo sem diminuir a taxa de transmissão e garantindo a transferência em “tempo real” dos dados.

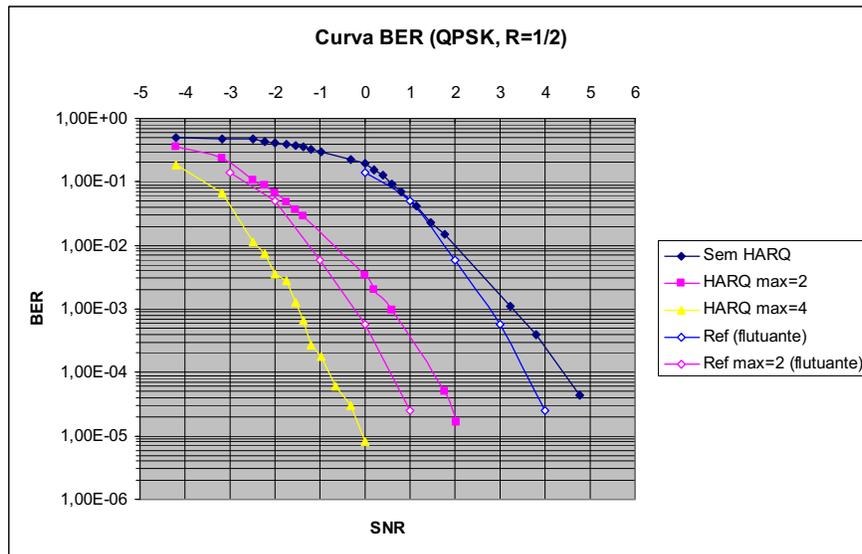


Figura 14: Curva de taxa de erro

O grande número de trocas entre o PC e a plataforma penaliza enormemente o tempo de execução, devido à baixa velocidade de transmissão da conexão Ethernet em relação à Rede-em-Chip. Além disso, o sistema não é independente. Um estudo mais detalhado de seu desempenho temporal é apresentado na seção “Análise dos Resultados”.

8 Novo Sistema

Para melhorar o desempenho temporal, todas as tarefas de controle da aplicação são embarcadas no sistema e gerenciadas pelo ARM (Tabela 4). O sistema torna-se então completamente autônomo do ponto de vista exterior. Com relação ao estudo precedente, os papéis de mestre e escravo são invertidos. O processador lê doravante os *flits* de configuração anteriormente armazenados nas memórias e os envia aos IPs via a interface “AHB \Leftrightarrow NoC”.

Tabela 4: Nova distribuição de tarefas entre o PC e o processador ARM

T A R E F A S	PC (escravo)	ARM (mestre)
		<i>Carregar as memórias</i>
	<i>Iniciar o processador e lançar a aplicação</i>	<i>Pedir os dados ao escravo</i>
	<i>Fornecer os dados na entrada</i>	<i>Configurar a cadeia de comunicação</i>
		<i>Iniciar o receptor</i>
		<i>Gerenciar as retransmissões</i>
		<i>Enviar ao PC os pacotes da saída</i>

As memórias embarcadas no sistema são repartidas segundo as necessidades da aplicação. As configurações iniciais, lançadas uma única vez quando da inicialização da aplicação, são armazenadas na memória externa, cujo acesso é mais lento. Durante a execução, as exigências de tempo tornam-se importantes e os dados envolvidos devem, portanto, ser rapidamente acessíveis. Por essa razão, a memória interna (RAM 2), ligada diretamente ao barramento AHB, é utilizada para os subprogramas do processador e para as configurações relacionadas aos roteiros “ACK” e “NACK”. A alocação das memórias foi cuidadosamente efetuada; os arquivos, as zonas e seus respectivos tamanhos e endereços são expostos nas Tabelas 5 e 6 e na Figura 15.

A rotina “*cfg_handler*” é encarregada de formatar os pacotes de *flits* conforme o protocolo NoC antes do envio deles na rede.

Tabela 5: Organização da memória externa

Arquivo	Tamanho (bits)	Número de Posições	Zona	Alocação (bits)	Endereços
<i>cfg_load_ext_ram_init.cfg</i>	5162x34	41296	INIT	1Mx8	0xC0000000 0xC00FFFFFF
<i>load_ext_ram_boot.cfg</i>	82x32	328	boot	2Kx8	0xC0100000 0xC01007FF

Tabela 6: Organização da memória RAM 2

Arquivo	Tamanho (bits)	Número de Posições	Zona	Alocação (bits)	Endereços
<i>load_sram_irq_handler.cfg</i>	158x32	632	IRQ_handler	2Kx8	0xB0000000 0xB00007FF
<i>cfg_load_sram_go_rx.cfg</i> <i>cfg_load_sram_read_x1_dma2.cfg</i>	27x34 20x34	108 80	GO_rx READ	512x8	0xB0000800 0xB00009FF
<i>cfg_load_sram_reset_fa_map.cfg</i>	5x34	20	RESET_map	256x8	0xB0000A00 0xB0000AFF
<i>cfg_load_sram_reset_rx_bit.cfg</i>	5x34	20	RESET_rx_bit	256x8	0xB0000B00 0xB0000BFF
<i>cfg_load_sram_ack.cfg</i>	249x34	1992	ACK	4Kx8	0xB0000C00 0xB0001BFF
<i>cfg_load_sram_nack.cfg</i>	220x34	1760	NACK	4Kx8	0xB0001C00 0xB0002BFF
<i>cfg_load_sram_ask_data.cfg</i>	40x34	320	ASK_data	1Kx8	0xB0002C00 0xB0002FFF
<i>load_sram_cfg_handler.cfg</i>	87x32	348	CFG_handler	1Kx8	0xB0003000 0xB00033FF
<i>load_sram_key_words.cfg</i>	10x32	40	KEY_words	256x8	0xB0003400 0xB00034FF

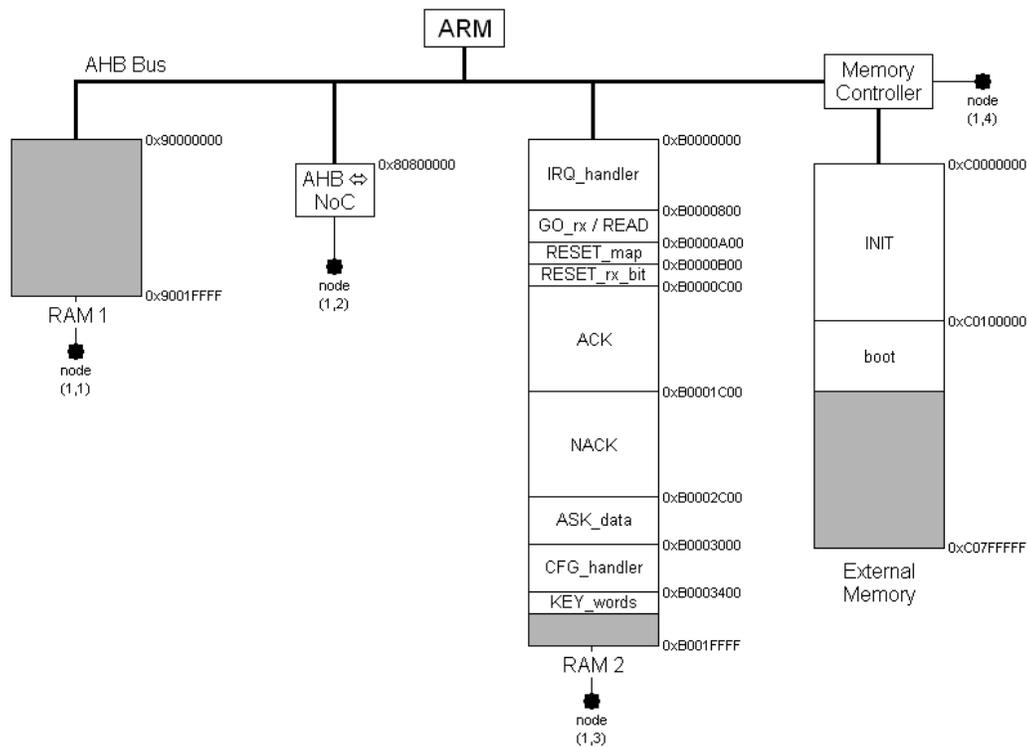


Figura 15: Disposição das memórias associadas ao processador ARM

Duas novas interrupções são introduzidas, uma para inicializar o receptor da cadeia de comunicação e a outra para pedir os dados ao PC. A primeira IT, que certifica a boa emissão de um pacote, provém do buffer 3 (x2_dma1) e serve sobretudo para a depuração. Ela pode ser igualmente útil para futuras implementações, por exemplo, a separação física entre o emissor, representando a estação de base, e o receptor, um terminal móvel.

Sabendo que o principal “retardador” do sistema é a interface Ethernet e como não era mais possível reduzir as trocas entre o PC e a plataforma, o carregamento “entrelaçado” de pacotes foi implementado. Enquanto um pacote ainda está em tratamento, o seguinte é pedido ao PC. Esses dois processos são portanto paralelos. O novo pacote é guardado no buffer 1 (x1_dma1), enquanto que o precedente pode ser recuperado do buffer 2 (f1_dma1) se uma retransmissão for solicitada. Assim, no caso de um ACK, toda a cadeia de comunicação (desde o primeiro buffer) entra em ação, enquanto que se um NACK acontece, a transmissão começa a partir do segundo buffer.

Dessa maneira, um mesmo pacote atravessa apenas uma única vez a conexão Ethernet, o que torna o sistema mais eficaz. A IT que pede os dados ao PC ocorre quando o buffer 2 é preenchido, permitindo que o conteúdo do buffer 1 seja substituído pelo pacote seguinte.

Os caminhos de roteamento (em anexo) foram também atualizados, já que as configurações são agora enviadas aos IPs pelo processador e não mais pelo PC.

O sistema é, de agora em diante, comandado pelo ARM. Este gerencia integralmente o protocolo HARQ e o PC tornou-se um simples fornecedor de dados. Os organogramas simplificados dos programas são expostos nas Figuras 16 e 17.

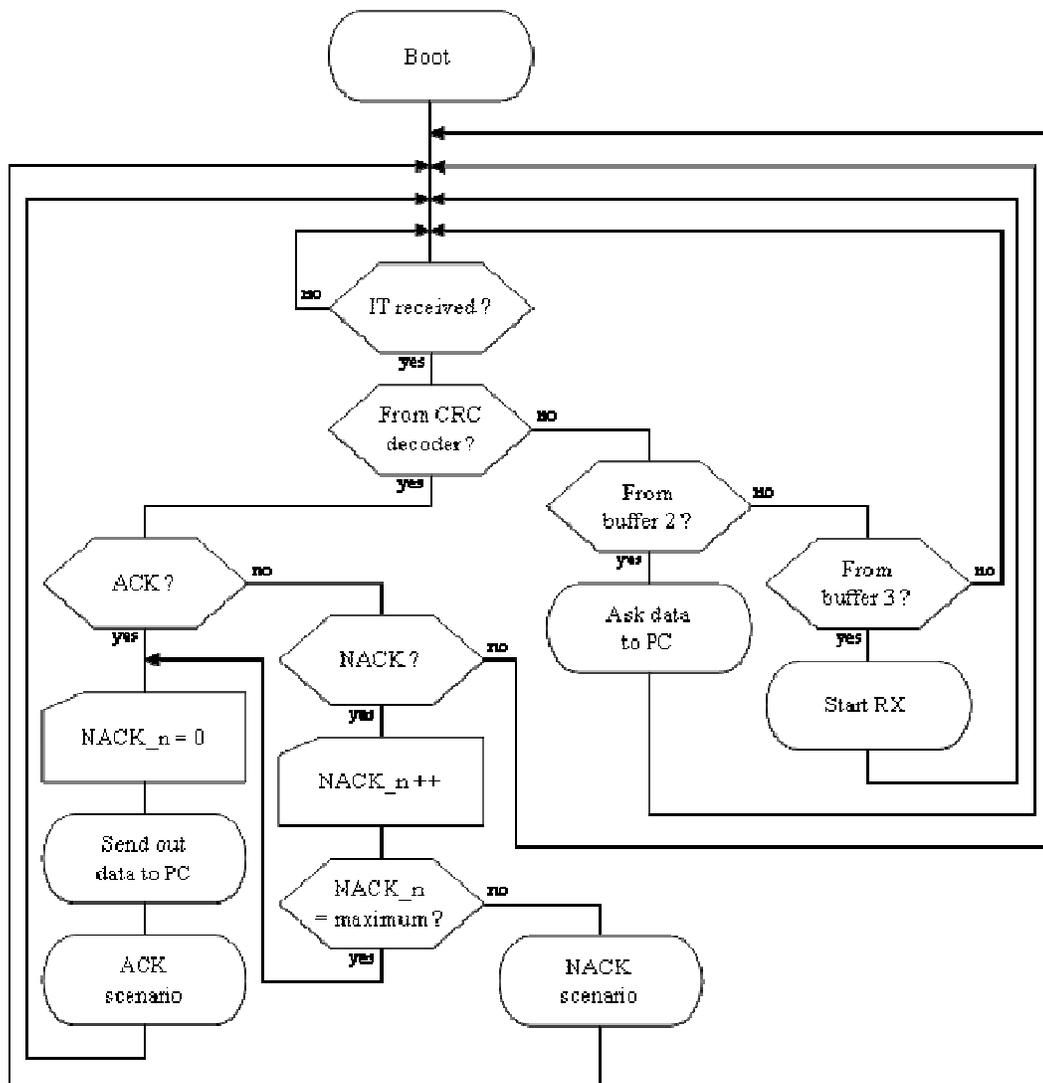


Figura 16: Organograma simplificado do novo programa do ARM

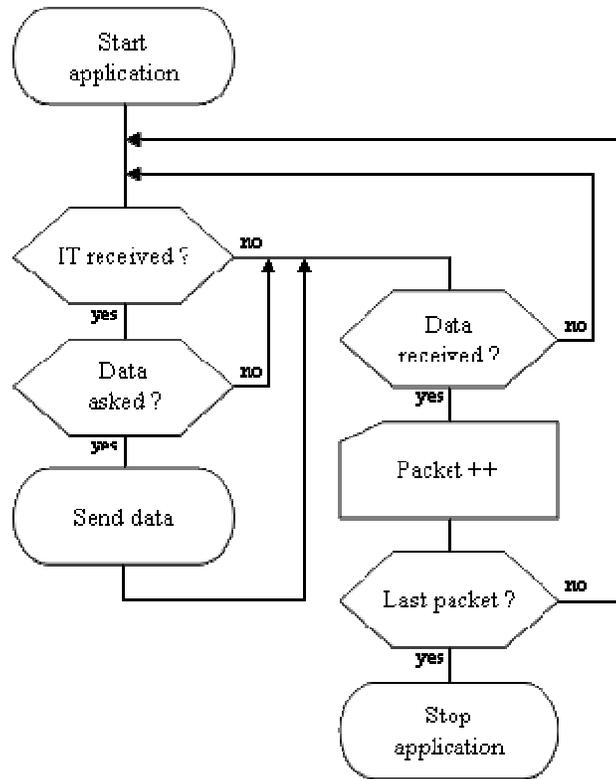


Figura 17: Organograma simplificado do novo programa do PC

Os diagramas de fluxo do novo sistema são mostrados nas Figuras 18, 19 e 20.

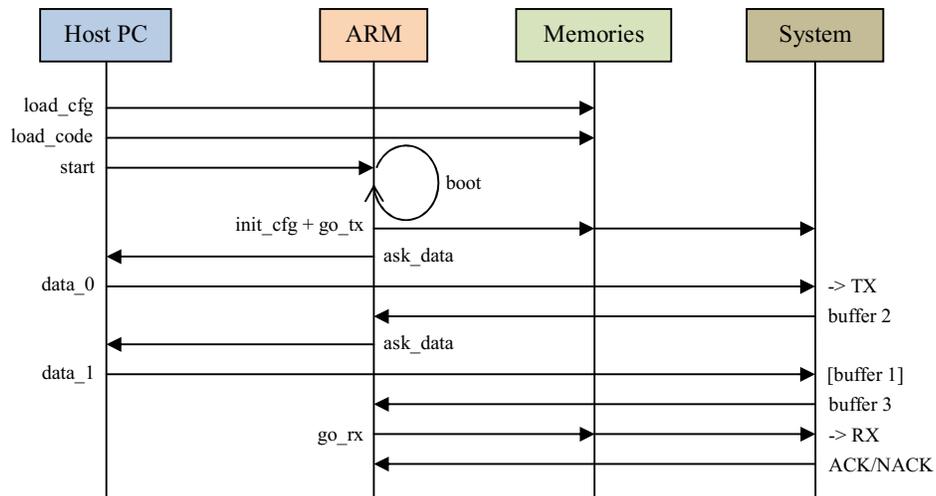


Figura 18: Novo diagrama de fluxo – inicialização

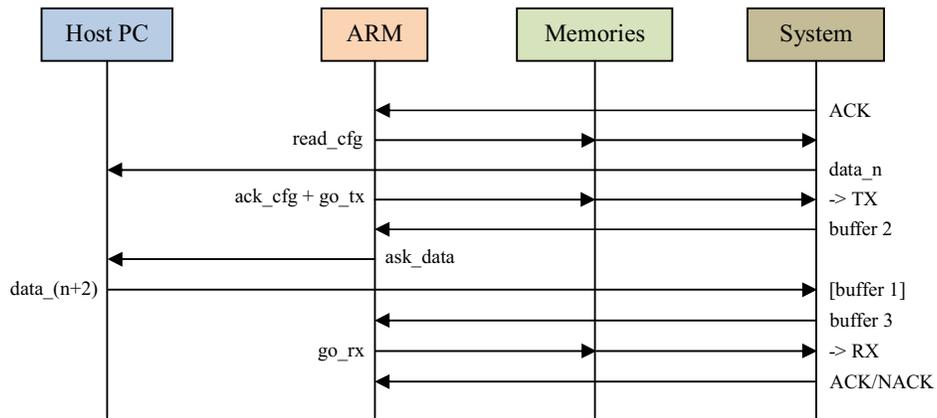


Figura 19: Novo diagrama de fluxo – “roteiro ACK”

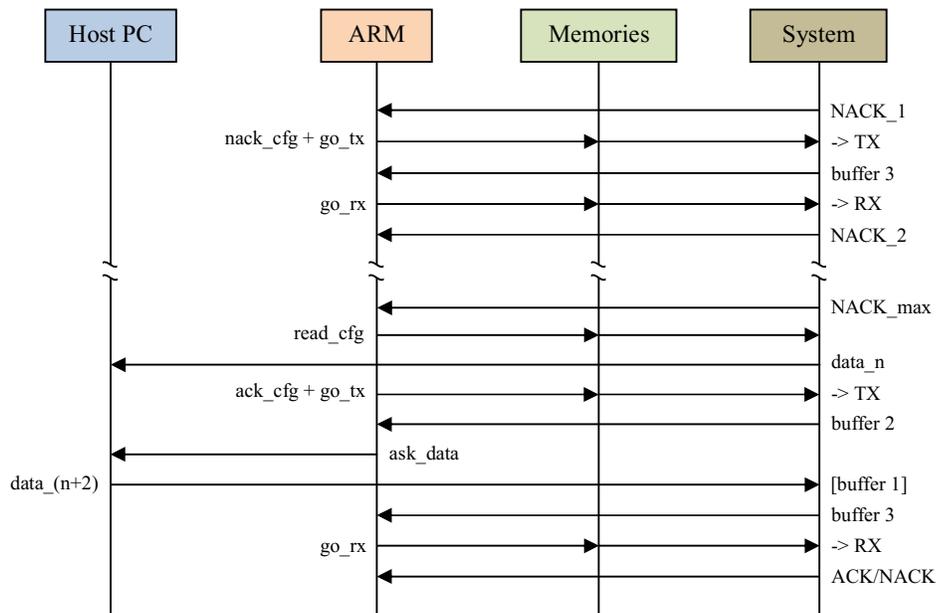


Figura 20: Novo diagrama de fluxo – “roteiro NACK”

A nova IHM, adaptada às novas necessidades, é mostrada na Figura 21.

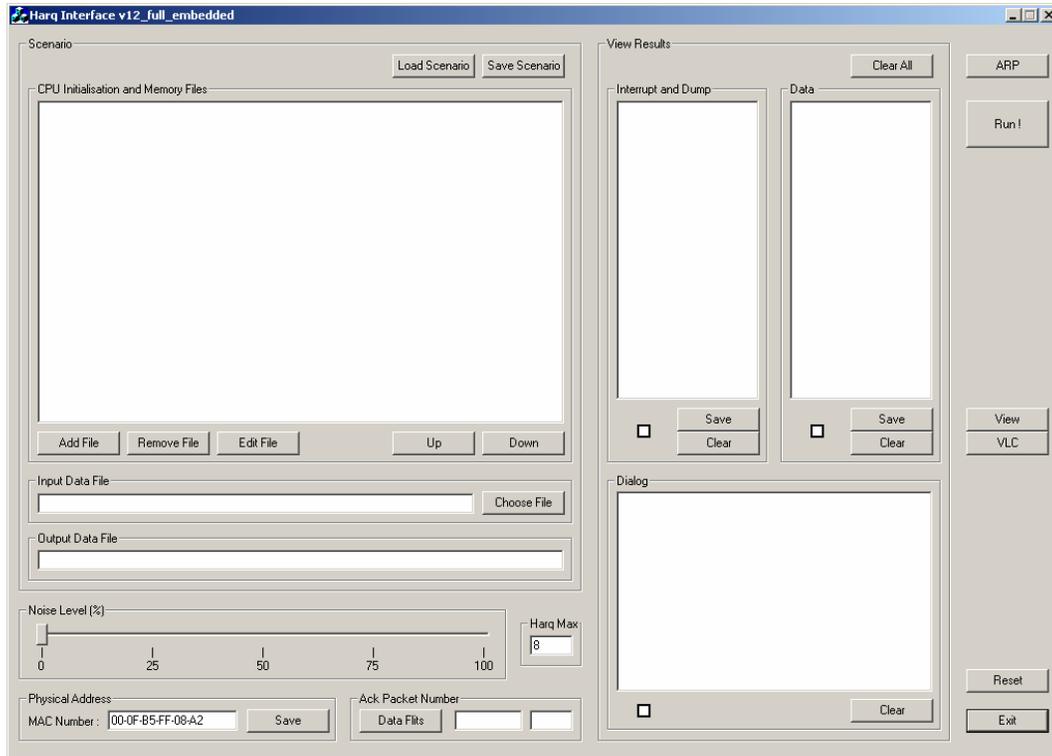


Figura 21: Interface homem-máquina do novo sistema

9 Análise dos Resultados

O novo sistema confirmou os resultados anteriores em termos de taxa de erro (Figura 22). O primeiro objetivo foi portanto alcançado, o de não comprometer a qualidade da transmissão em benefício de um melhor tempo de execução.

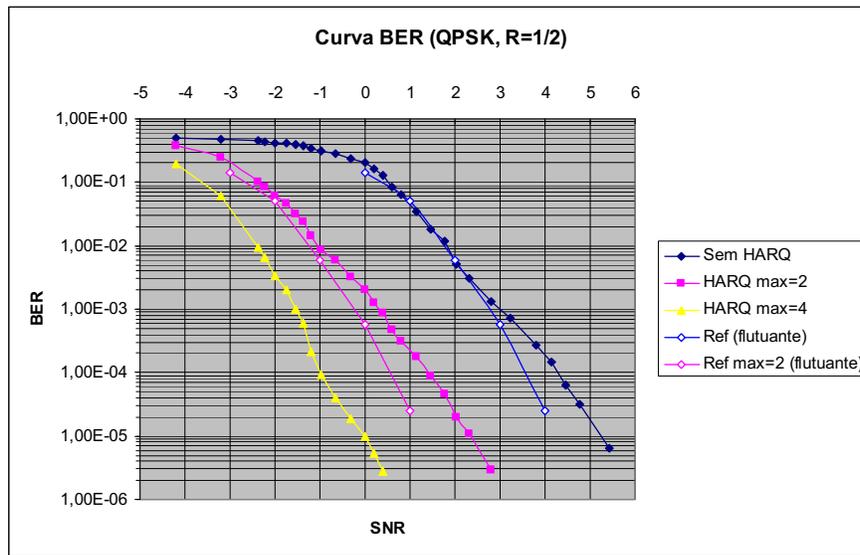


Figura 22: Nova curva de taxa de erro

A fim de comparar o desempenho temporal dos dois sistemas, a transmissão de um arquivo de vídeo (Tabela 7) foi efetuada para vários níveis de ruído do canal e variando também o número máximo de retransmissões possíveis, de forma a haver sempre uma recepção sem erros. Os resultados são apresentados nas Tabelas 8 e 9 e nas Figuras 23, 24 e 25.

Tabela 7: Arquivo de referência para as medições de tempo de transmissão

Arquivo de Referência	Tamanho	Nº de Pacotes	Taxa CBR	Duração Efetiva
<i>pub_scrabble.mpg</i>	7405600 bits	15961	400 Kbits/s	15s

Tabela 8: Tempo de transmissão do sistema existente

Nível de Ruído		Nº Máximo de Retransmissões	Tempo de Transmissão
0%	-	0	24,5 min
21%	4,45dB	1	25,3 min
25%	2,79dB	2	26,7 min
30%	1,14dB	3	41,5 min
35%	0dB	4	47,8 min

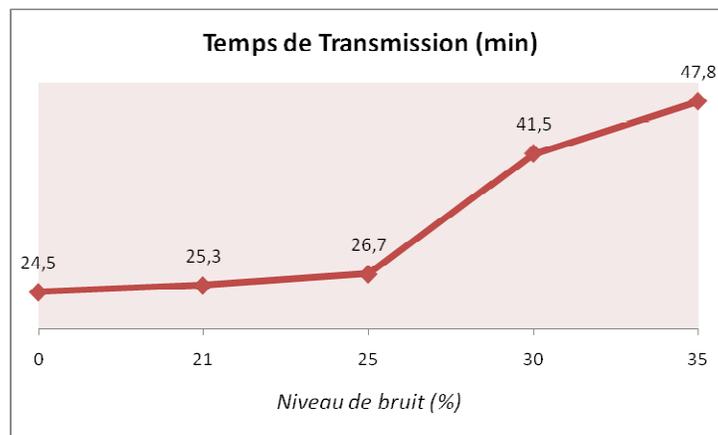


Figura 23: Tempo de transmissão do sistema existente

Tabela 9: Tempo de transmissão do novo sistema

Nível de Ruído		Nº Máximo de Retransmissões	Tempo de Transmissão
0%	-	0	11,5s
21%	4,45dB	1	11,5s
25%	2,79dB	2	11,5s
30%	1,14dB	3	11,6s
35%	0dB	4	11,6s

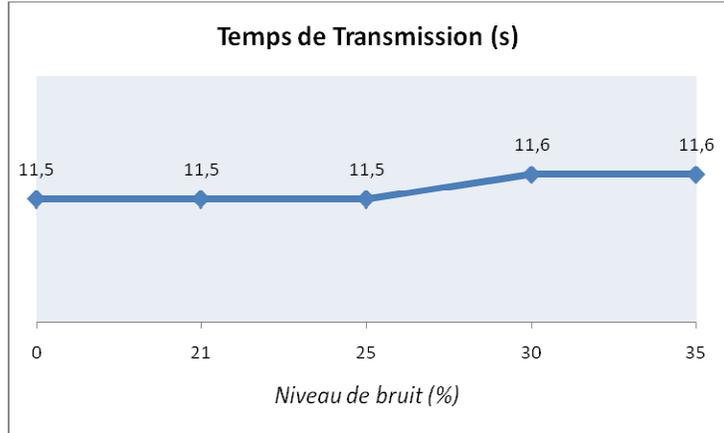


Figura 24: Tempo de transmissão do novo sistema

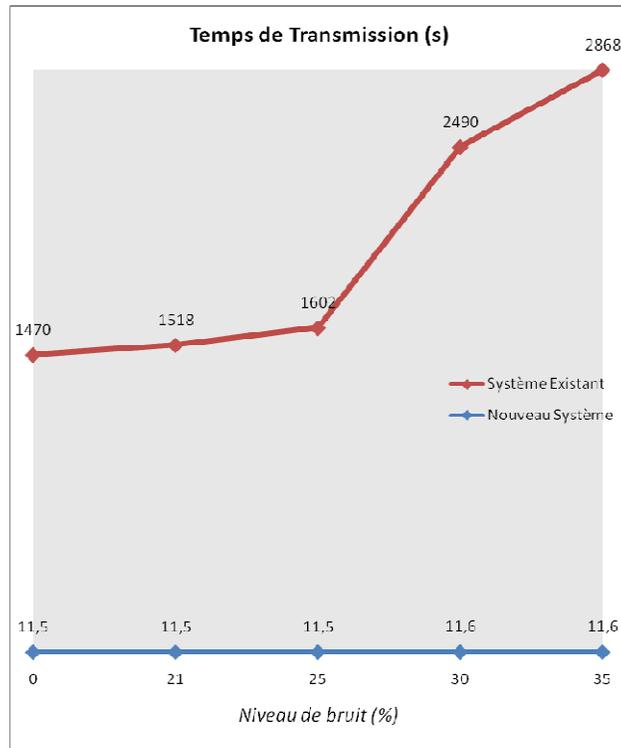


Figura 25: Tempo de transmissão dos dois sistemas

Visivelmente na Figura 25, uma enorme diferença entre os dois sistemas foi constatada. O ganho de desempenho é de um fator superior a 120 no pior dos casos. Outro fenômeno observável é a sensibilidade do tempo de execução ao número de retransmissões. Enquanto o sistema existente se mostrou fortemente influenciável, o tempo de transmissão do novo sistema praticamente não mudou para as diferentes configurações, o que é extremamente positivo.

Os resultados obtidos foram portanto largamente satisfatórios e permitem a evolução do sistema para um funcionamento “tempo real” (os tempos medidos são inferiores à duração efetiva do vídeo).

Uma vez minimizadas as trocas com o PC e paralelizada a aplicação, os novos limites de desempenho residem nas capacidades dos materiais que constituem o sistema.

10 Conclusão

A expansão do campo das telecomunicações demanda o aparecimento de novos sistemas, a fim de responder às suas necessidades crescentes: a compatibilidade entre múltiplos padrões e taxas de transferência cada vez mais elevadas.

Revela-se assim essencial implementar diferentes protocolos em uma única plataforma física capaz de tudo gerenciar.

As Redes-em-Chip parecem ser uma excelente escolha para satisfazer às novas exigências. Elas permitem a integração fácil de unidades anteriormente desenvolvidas e a reconfiguração dinâmica delas.

A plataforma colocada à disposição, que dispõe de uma Rede-em-Chip, permite validar com flexibilidade a realização de novas normas de comunicação.

A migração da cadeia de comunicação estudada para o microsistema permitiu conduzi-la a desempenhos em “tempo real” sem prejudicar a qualidade da transmissão. A totalidade das tarefas foi transferida do PC para o processador ARM embarcado, o que tornou a aplicação autônoma.

A sequência do desenvolvimento efetuado permitiria a implementação de um verdadeiro sistema em “tempo real” e mais adaptado ao contexto das comunicações sem fio. Um potencial tema de pesquisa diz respeito à instanciação de vários processos HARQ “Stop and Wait” concorrentes em diferentes canais, a fim de compensar a dependência ao tempo de espera da sinalização.

Bibliografia

- CEA. Rapport Annuel 2007;
- M. Marcaud, F. Perrin. DCIS et DTBS: “Livret d’accueil du nouvel arrivant”, março de 2008;
- S. Min. Rapport de projet de troisième année: “Etude en vue d’implémentation d’un algorithme de transmission par paquets sur le processeur ARM d’un ASIC de type Network on Chip”, junho de 2006;
- S. Min. Rapport de DRT: “Intégration d’un processus d’acquittement de paquets sur une plate-forme matérielle à base de réseau sur puce dédiée à des applications de communication sans fil”, julho de 2007;
- I. Horton (traduction de l’anglais par D. Ulivi). Visual C++ 6, 1999.

Anexo: Caminhos de Roteamento

Tabela 10: Caminhos de roteamento

IP	A partir do PC		A partir do processador ARM	
	→	←	→	←
Buffer 1 (x1_dma1)	<i>NWN</i>	<i>0000C</i>	<i>WSSSWNNN</i>	<i>003AB</i>
Codificador CRC (x1)	<i>NWSW</i>	<i>000EC</i>	<i>WNNNWSW</i>	<i>03B03</i>
Buffer 2 (f1_dma1)	<i>NESN</i>	<i>00024</i>	<i>WNNNESN</i>	<i>00903</i>
Codificador Convolutcional (f1)	<i>NEEEEE</i>	<i>00554</i>	<i>WNNNEEEEE</i>	<i>15503</i>
Entrelaçador (f1)	<i>NEEEEW</i>	<i>00D54</i>	<i>WNNNEEEEW</i>	<i>35503</i>
Mapeador (f1)	<i>NEEEW</i>	<i>00354</i>	<i>WNNNEEEW</i>	<i>0D503</i>
Buffer 3 (x2_dma1)	<i>NNWN</i>	<i>00030</i>	<i>WNNSSWNNN</i>	<i>00E83</i>
Canal AWGN (x2)	<i>SSWNS</i>	<i>0023A</i>	<i>WNNWSN</i>	<i>002C3</i>
Buffer 4 (x2_dma2)	<i>NNWE</i>	<i>00070</i>	<i>WNNSSWNNNS</i>	<i>20E83</i>
Buffer LLR (f2_dma1)	<i>SENNS</i>	<i>00206</i>	<i>NWSN</i>	<i>0002C</i>
Demapeador (f2)	<i>SEEEW</i>	<i>00356</i>	<i>ESSN</i>	<i>00029</i>
Desentrelaçador (f2)	<i>SEEEEW</i>	<i>00D56</i>	<i>EESSN</i>	<i>000A5</i>
Combinador HARQ (x2)	<i>SSNNS</i>	<i>0020A</i>	<i>WNNE</i>	<i>00043</i>
Decodificador Convolutcional (f2)	<i>SENEEW</i>	<i>00D46</i>	<i>ESN</i>	<i>00009</i>
Decodificador CRC (x2)	<i>SSWNE</i>	<i>0013A</i>	<i>WNNWSE</i>	<i>006C3</i>
Buffer 5 (x1_dma2)	<i>NWE</i>	<i>0001C</i>	<i>WSSSWNNNS</i>	<i>083AB</i>
ETH (x1)	<i>W</i>	<i>00003</i>	<i>WSSSNW</i>	<i>00CAB</i>