

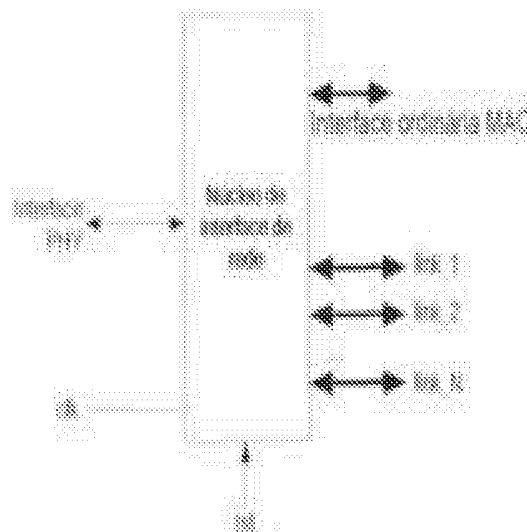
(12) **FASCÍCULO DE PATENTE DE INVENÇÃO**

(22) Data de pedido: <b>2007.05.16</b>	(73) Titular(es): <b>COREWORKS, S.A.</b>
(30) Prioridade(s):	<b>RUA ALVES REDOL, N.º 9 1000-029 LISBOA PT</b>
(43) Data de publicação do pedido: <b>2008.11.17</b>	(72) Inventor(es):
(45) Data e BPI da concessão: /	<b>JOSÉ TEIXEIRA DE SOUSA PT</b> <b>NUNO CALADO CORREIA LOURENÇO PT</b> <b>NELSON GONÇALO DO ROSÁRIO RIBEIRO PT</b> <b>VICTOR MANUEL GONÇALVES MARTINS PT</b> <b>RICARDO JORGE SANTOS MARTINS PT</b>
	(74) Mandatário: <b>MARIA SILVINA VIEIRA PEREIRA FERREIRA</b> <b>RUA CASTILHO, N.º 50, 5º - ANDAR 1269-163 LISBOA PT</b>

(54) Epígrafe: **ARQUITECTURA DE ACESSO AO NÚCLEO DE REDE.**

(57) Resumo:

A arquitectura proposta encontra-se integrada num sistema em pastilha (SoC) genérico e consiste numa interface de rede expandida e numa infra-estrutura para o acesso aos núcleos de Propriedade Intelectual (IP) no sistema. A arquitectura permite que o sistema em pastilha comunique com um posto de trabalho de um utilizador, que esteja ligado a uma rede de comunicação. A invenção pode ser utilizada como uma interface de rede simplificada para troca de dados, não exigindo processadores integrados e respectivos softwares. A arquitectura pode ser utilizada como estímulo e para observar as respostas durante a fase de desenvolvimento de um núcleo IP, através da utilização de uma estação de protótipos de Rede de Portas Lógicas Programáveis (FPGA). A arquitectura pode ser usada para programar processadores integrados em toda a rede, ou para configurar unidades lógicas programáveis integradas, tais como as FPGAs integradas. A invenção pode ser usada para substituir temporariamente a normal entrada e saída de dados de um núcleo IP, com estímulos e respostas usados para várias finalidades: verificação funcional; substituição da miríade de protocolos seriais e paralelos existentes para vários fins, contribuindo para a unificação de métodos entrada/saída; testar, i.e., detectar a presença de defeitos de fabrico físicos; impulsionar e analisar a presença de erros de design (bugs); diagnosticar e solucionar problemas no campo; actualizar remotamente as configurações e versões de suporte lógico inalterável de sistemas já utilizados. A arquitectura implementa o acesso paralelo a núcleos IP, formando uma árvore de acesso paralelo no sistema, enraizada no núcleo da interface de rede. A invenção facilita funções básicas de software com base em protocolos de



comunicação padrão, os quais são executados no posto de trabalho do utilizador e são usados para a gestão da infra-estrutura de acesso ao núcleo e para comunicar com os núcleos IP individuais do sistema em pastilha. Ao proporcionar uma interface de rede, todas as actividades acima enumeradas podem ser desempenhadas de forma cooperante, permitindo que equipas, geograficamente distribuídas, realizem simultaneamente os seus diferentes desempenhos.

## RESUMO

### "ARQUITECTURA DE ACESSO A NÚCLEOS POR REDE"

A arquitectura proposta é integrada num sistema integrado (SI) e consiste numa interface de rede integrada e uma infra-estrutura para acesso a núcleos de propriedade intelectual (NPI), dentro do sistema. A arquitectura permite que o SI comunique com uma estação de trabalho de um utilizador ligado a uma rede de comunicações. A invenção pode ser usada como uma interface de rede simplificada para troca de dados, a qual não requer processadores embebidos e respectivo software. A arquitectura pode ser usada para fornecer estímulos e observar respostas durante a fase de desenvolvimento de um NPI utilizando uma estação de prototipagem baseada em Matriz de Portas Programáveis no Terreno (MPPT). A arquitectura pode ser usada para programar processadores embebidos pela rede ou configurar dispositivos como MPPTs embebidos. A invenção pode ser usada para substituir temporariamente a entrada e saída de dados de um NPI com estímulos e respostas utilizadas para vários fins: verificação funcional; substituir a miríade de protocolos série e paralelo que existem para vários propósitos, contribuindo para unificar métodos de entrada/saída; teste, isto é, detectar a presença de defeitos físicos de fabrico; exercitar e detectar a presença de erros de projecto; diagnosticar e reparar falhas no terreno; actualizar remotamente a configuração e versões dos programas embebidos de sistemas já em funcionamento. A arquitectura implementa acesso paralelo a NPIs, formando uma árvore de acesso paralelo no sistema, com raiz no núcleo de interface de rede.

A invenção providencia funções básicas de software baseadas em protocolos de comunicação normalizados, as quais são

executadas numa estação de trabalho do utilizador e são utilizadas para gerir a infra-estrutura de acesso a NPIs e para comunicar individualmente com os NPIs no SI. Providenciando uma interface de rede, todas as actividades descritas acima podem ser levadas a cabo de forma colaborante, permitindo o trabalho simultâneo de equipas geograficamente distribuídas.

## DESCRIÇÃO

### "ARQUITECTURA DE ACESSO A NÚCLEOS POR REDE"

#### Campo da Invenção

Esta invenção está relacionada na generalidade com o projecto, verificação e teste de sistemas integrados (SI) ou, em inglês, Systems on a Chip (SoCs), baseados em núcleos de propriedade intelectual (NPI) ou, em inglês, IP cores, e em particular ao acesso a NPIS para entrada e saída de dados, configurações, controlo e observação, através de redes de comunicação multi-utilizador.

#### Antecedentes da invenção

Um circuito integrado que implementa um SI complexo necessita de ser acedido para operações funcionais e não funcionais por inúmeras razões, nomeadamente: configuração a partir do exterior, observação e controlo, manutenção e actualizações.

Os SIs tem de se adaptar às mais variadas circunstâncias externas, ambientes envolventes e normas existentes. Para conseguir esta flexibilidade, registos de configuração são usados para guardar valores de configuração que influenciam o comportamento dos NPIS. Adicionalmente, é muitas vezes necessário emitir comandos durante o funcionamento do NPI e monitorar o seu estado; isto é normalmente conseguido à custa de registos de controlo e de estado. Informação detalhada sobre o estado do sistema ou programação extensa pode ser obtida descarregando ou carregando todo o conteúdo de memórias internas.

Esquemas anteriores foram propostos para acessar a circuitos integrados para controle, observação e configuração. Estas funções tendem a ser separadas de outras funções de comunicação e de usar pinos específicos. Os esquemas tradicionais são normalmente protocolos série de baixa velocidade, os quais evitam usar muitos pinos do SI. Exemplos bem conhecidos de esquemas desses são o barramento I<sup>2</sup>C da Philips [1] ou o barramento SPI da Motorola [2]. O standard JTAG, inicialmente desenvolvido para fins de teste, também se tornou um dos principais métodos para configuração e verificação, e constitui a norma IEEE 1149.1 [3].

As Matrizes de Portas Programáveis no Terreno (MPPT) ou, em inglês, Field Programmable Gate Array (FPGA), utilizam o JTAG para configuração do SI. Sendo as MPPTs totalmente configuráveis, necessitam de quantidades massivas de dados de configuração, os quais, quando introduzidos por JTAG, levam um tempo longo a carregar no SI.

Os fornecedores de MPPTs desenvolveram métodos de configuração diferentes do JTAG de modo a reduzir o tempo de configuração. Estes esquemas usam normalmente uma interface paralela e carregam os dados de configuração de uma memória externa não volátil, o que é menos flexível do que obter os dados de configuração de um computador, como acontece com o JTAG.

Uma dificuldade conhecida ao utilizar JTAG é quando apenas parte da MPPT necessita de ser reconfigurada. Para resolver este problema os fornecedores de MPPTs oferecem esquemas de reconfiguração parcial que dividem a MPPT em regiões e permitem a reconfiguração individual de cada região.

Contudo, estas regiões devem ser amplas ou de outro modo o encargo com o tempo de preparação da configuração fica demasiado dispendioso. Por outro lado, o I<sup>2</sup>C e o SPI podem ser usados para ler ou escrever pequenos registos de configuração, mas falta-lhes a habilidade para lidar com grandes quantidades de dados de reconfiguração. Uma característica ainda mais avançada é a possibilidade de reconfiguração durante a execução de programas. Em algumas aplicações, tem-se verificado que é vantajosa a reconfiguração total ou parcial do circuito integrado enquanto o programa está em execução. Isto rentabiliza melhor a lógica disponível, poupa área e melhora o desempenho. A infra-estrutura necessária para a reconfiguração em execução tem de ser suficientemente rápida para que não tenha impacto no tempo de computação. Acresce, que a reconfiguração deve ocorrer sincronamente com o relógio de sistema, o que é difícil de garantir dado que as técnicas habituais de reconfiguração usam o seu próprio relógio. As técnicas habituais de configuração são desadequadas para este fim, dado que são em série, lentas e precisam de relógios dedicados, diferentes do relógio do sistema.

Nas técnicas de configuração tradicionais, um dispositivo configurável pode ser programado pela rede se estiver ligado a um computador que por sua vez está ligado à rede. Porém, como as técnicas de configuração tradicionais são lentas, qualquer função que exija que dados sejam enviados ou recebidos a alta velocidade, a partir do dispositivo, não pode ser executada. Neste conjunto de funções incluem-se a verificação, inspecção para procura de erros, teste,

reconfiguração parcial ou total, rápida, e enquanto o programa está em execução.

As técnicas referidas têm como objectivo aceder a circuitos integrados. Com a densidade de integração possível hoje em dia, os circuitos impressos têm gradualmente vindo a ser substituídos por SIs, onde os circuitos integrados são agora substituídos por blocos funcionais, todos embebidos e integrados num único cristal de silício. Assim, em vez de acedermos a circuitos integrados independentes, surge a necessidade de aceder a blocos no interior dum sistema integrado, SI. A norma IEEE 1500, extensão da norma IEEE 1149, propõe um método de aceder a blocos embebidos num sistema integrado, porém apenas para teste. Esta norma tem a limitação de implementar um protocolo série, o que implica a existência de registos de leitura longos dentro do circuito integrado (SI). A consequência é lentidão tanto da aplicação dos estímulos como da leitura dos resultados. Por outro lado, os registos de leitura não replicam a natureza hierárquica da estrutura dos circuitos digitais, sendo necessário os mecanismos previstos na patente US2003/120986, para que um funcionamento hierárquico seja possível.

A invenção agora apresentada endereça todos os problemas referidos, embebendo estruturas de controlo, observação e configuração na estrutura de comunicação do próprio circuito integrado, e usando para isso uma interface de rede normalizada no interior do próprio sistema. Deste modo a comunicação para o exterior do circuito integrado é série e de alta velocidade, porém a comunicação interna entre blocos é paralela, de baixa velocidade e em mapa de memória. Usando a técnica proposta, a controlabilidade,

observabilidade e configuração do sistema integrado é conseguida sem a necessidade de equipamento externo adicional como fichas e cabos JTAG. Consegue-se assim uma taxa de transferência de dados semelhante à duma rede de comunicação, o que significa uma largura de banda muito superior às conferidas pelas técnicas de acesso tradicionais. Esta técnica facilita também a reconfiguração dado que é hierárquica e dado que os registos internos são endereçáveis. A reconfiguração com o programa em execução é também simplificada dado que todo o sistema é síncrono. Com esta invenção os diversos NPIs dentro dum sistema integrado, ficam interligados com uma rede. Assim, estes blocos podem ser directamente acedidos para verificação, teste, manutenção e actualização.

#### **Resumo da invenção**

A arquitectura proposta está integrada num Sistema Integrado (SI) genérico e consiste numa interface de rede expandida e uma infra-estrutura de acesso aos NPIs no sistema. Serve para auxiliar o desenvolvimento de novos núcleos ou à integração de blocos existentes, programação e configuração de blocos no sistema, verificação da funcionalidade de NPIs, teste de NPIs, inspecção para procura de erros em NPIs, manutenção e actualização. A presente invenção pode também ser usada como uma interface de rede simplificada para transferência de dados, que não precisa dum processador embebido e do respectivo programa. Pelo facto de se fornecer uma interface com infra-estrutura de rede, todas as actividades enumeradas anteriormente podem ser realizadas duma forma colaborante, permitindo que equipas dispersas geograficamente possam realizar as suas tarefas em simultâneo. A arquitectura pode ainda ser usada para gerar estímulos e observar resultados durante a fase

de desenvolvimento dum bloco de IP usando placas de prototipagem com FPGAs. A arquitectura pode ser usada para programar processadores embebidos através de uma rede ou para configurar dispositivos programáveis embebidos tais como FPGAs embebidas. A arquitectura pode ser usada para configurar vários registos de controlo ou observar vários registos de estado do sistema. O invento pode ser usado para temporariamente substituir a entrada e saída de dados normal de um NPI por estímulos e respostas usadas para verificação funcional. A arquitectura pode ser usada para substituir a miríade de protocolos série e paralelo existentes com diferentes finalidades, contribuindo para unificar as interfaces entrada/saída. O invento pode ser usado para temporariamente substituir a entrada e saída de dados normal de um núcleo por estímulos e respostas usadas para teste, i.e., para detectar a presença de defeitos físicos que ocorram na fabricação. O invento pode ser usado para temporariamente substituir a entrada e saída de dados normal de um núcleo por estímulos e respostas usadas para estimular e detectar a presença de erros de projecto. O invento pode ser usado para temporariamente substituir a entrada e saída de dados normal de um núcleo por estímulos e respostas usadas para diagnóstico e correcção de problemas que ocorram durante a sua operação. A arquitectura pode ser usada para actualizar remotamente as versões de configuração e de firmware de sistemas em funcionamento. A arquitectura implementa acesso paralelo a núcleos, formando uma árvore de acesso paralelo no sistema, com raiz no núcleo de interface de rede. A arquitectura permite que o SI comunique com uma estação de trabalho ligada à rede de comunicações. O invento disponibiliza funções básicas de software para a estação de trabalho baseadas em protocolos de comunicação normalizados para

gestão da infra-estrutura de acesso ao núcleo e para comunicação com os restantes núcleos existentes no SI.

### **Breve descrição das figuras**

Fig. 1- Panorâmica da Tecnologia de Acesso a Núcleos por Rede

Fig. 2- Hierarquia de acesso ao núcleo

Fig. 3- Interface externa simplificada do núcleo de interface de rede

Fig. 4- Diagrama de blocos do núcleo de interface de rede

Fig. 5- Detalhe da interface entre a núcleo de interface de rede e os núcleos do sistema

Fig. 6- Estrutura preferencial para uma palavra de ligação

Fig. 7- Detalhe da interface preferencial de um núcleo do utilizador para o núcleo de interface de rede

Fig. 8- Acesso aos registos dos núcleos filhos

Fig. 9- Pilha do protocolo de rede para a realização preferencial

Fig. 10- Formato dos pacotes CADP

### **Descrição detalhada da realização preferencial**

A Fig. 1 representa uma perspectiva geral do invento, o qual consiste em componentes de hardware e software. Os

componentes de hardware são aplicados a um SI do utilizador e consistem numa interface dedicada designada núcleo de interface de rede (NIR) e numa árvore de acesso a núcleos. A componente de software corre numa estação de trabalho do utilizador e consiste numa biblioteca de funções para acesso ao NIR, aplicações de software específicas para cada núcleo, as quais devem ser integradas numa aplicação do sistema.

Este invento permite que os núcleos embebidos num SI, não necessariamente processadores embebidos, sejam acedidos através de uma rede de comunicações. Na realização preferencial será utilizada a Ethernet, mas este invento aplica-se a outros tipos de redes, desde que um núcleo de interface de rede e aplicações de software equivalentes sejam desenvolvidos para o tipo de rede em questão.

Durante o normal funcionamento o NIR pode ser usado como se fosse um núcleo de controlo do acesso ao meio, em inglês, Medium Access Control (MAC), ligado a um processador embebido, utilizando um interface MAC (ver Fig. 1). O processador embebido normalmente implementa as camadas superiores da pilha de protocolos: IP (Internet Protocol), TCP (Transport Control Protocol), UDP (User Datagram Protocol), etc.

Num modo especial de funcionamento, o modo de acesso a núcleos, o NIR comunica com núcleos no SI.

Na Fig. 1, os núcleos IP1, IP3 e IP4 estão ligados directamente ao NIR. Existem N pares de ligações directamente ligados aos núcleos individuais, onde o par de ligações  $i$  tem a designação  $link\_out(i)$  e  $link\_in(i)$ . O

NIR tem igualmente uma interface com o núcleo ou o chip externo responsável pela implementação da camada física do protocolo - tal como o dispositivo PHY que se pode ver na Fig. 1.

O NIR é acedido remotamente através de um computador do utilizador, onde o controlador de software da arquitectura está instalado. O controlador é implementado no topo do protocolo UDP, usando comunicação por *sockets*. As funções mais básicas do controlador de software permitem enviar/receber pacotes de informação para/de um núcleo em particular no SI, incluindo o próprio NIR.

Alguns núcleos podem comunicar indirectamente com o NIR. Na Fig. 1, o núcleo IP2 está indirectamente ligado ao NIR através do núcleo IP1. O esquema de acesso é recursivo formando um acesso em árvore: núcleos directamente ligados ao NIR são denominados núcleos de nível 1; núcleos ligados a núcleos de nível 1 são de nível 2, na generalidade núcleos ligados a núcleos de nível  $j$  são núcleos de nível  $j+1$ . Preferencialmente, a ligação entre um núcleo do nível  $j$  e os núcleos no nível  $j+1$  devem imitar a ligação entre o NIR e os núcleos de nível 1. Na Fig. 2 está representada a árvore de acesso.

A interface externa do NIR é representada na Fig. 3 e um diagrama simplificado do NIR está representado na Fig. 4.

Quer no acesso em modo normal quer no acesso aos núcleos, os blocos Ethernet e ARP (Address Resolution Protocol), tanto receptor (RX) como transmissor (TX) são utilizados.

Durante o normal funcionamento a interface habitual de MAC está activa e os pacotes Ethernet são encaminhados por norma a um processador embebido, o qual implementa as camadas superiores da pilha de protocolos: UDP/IP, TCP/IP, etc.

No modo de acesso a núcleos, a interface MAC está desactivada e os protocolos de camadas superiores são efectuados a nível de hardware. São fornecidos, na realização preferencial, descrições dos blocos de hardware para versões simplificadas dos protocolos IP e UDP.

A invenção adiciona uma camada particular à pilha de camadas, denominada Core Access Datagram Protocol (CADP), a qual é implementada pelos blocos CADP RX e TX conforme se mostra na Fig. 4. O bloco CADP RX recebe informação da rede e fornece-se aos núcleos. O bloco CADP TX passa informação recebida dos núcleos à rede. Seguidamente faz-se uma descrição sistemática de cada bloco da Fig. 4:

**Ethernet RX:** Módulo da camada de ligação que permite receber informação utilizando o protocolo MAC e a extracção do ARP ou os pacotes IP dentro dos pacotes MAC. Internamente verifica a integridade dos dados através do CRC (*Cyclic Redundancy Check*).

**Ethernet TX:** Módulo de ligação que permite a transmissão de informação utilizando o protocolo MAC através do encapsulamento de pacotes ARP ou IP adicionando cabeçalhos MAC. Internamente calcula o CRC e anexa-o ao endereço do pacote enviado.

**ARP RX:** Módulo da camada de ligação que permite receber mensagens utilizando ARP e detectar o tipo de mensagem (pedido ARP ou resposta ARP). Internamente este módulo mantém uma tabela de 2 endereços de Ethernet indexados pelo endereço de IP, a qual é consultada pelo módulo de ARP TX.

**ARP TX:** Módulo da camada de ligação que permite enviar mensagens (pedido ARP ou resposta ARP) utilizando o protocolo ARP. Estas mensagens são enviadas a pedido dos módulos ARP RX (mensagens de resposta ARP) e IP TX (mensagens de pedidos ARP).

**IP RX:** Módulo da camada de rede que permite receber pacotes de dados IP desencapsulando os pacotes UDP enviados como carregamentos de pacotes IP. Quaisquer outros tipos de carregamentos são ignorados.

**IP TX:** Módulo de bloco de rede que permite enviar pacotes de dados IP inserindo cabeçalhos IP nos pacotes UDP provenientes do módulo UDP TX.

**UDP RX:** Módulo da camada de transporte que permite receber dados através do protocolo UDP, extraíndo os pacotes CACP desses dados. Estes pacotes são reencaminhados para o módulo CACP RX responsável pelas últimas operações na recepção de pacotes.

**UDP TX:** Módulo da camada de transporte que permite enviar dados CACP encapsulados em pacotes UDP, através da inserção de cabeçalhos UDP nos pacotes vindos do módulo CACP TX.

**CADP RX:** Recebe pacotes CADP de dados e controlo vindos do módulo UDP RX. Este módulo analisa os campos CADP e faz o seguinte: (1) envia os dados acabados de receber para um dos interfaces *link\_out\_i* e pede ao modulo CADP TX para enviar o pacote que confirmação respectivo; (2) descarta e reenvia a confirmação de pacotes retransmitidos; (3) informa o módulo CADP TX das confirmações recebidas; (4) reage a comandos CADP.

**CADP TX:** Envia pacotes CADP de dados e controlo para o módulo UDP TX. Este módulo faz o seguinte: (1) recebe os dados das interfaces para os NPI e insere-os em pacotes CADP de até 240 palavras; (2) envia pacotes CADP de confirmação após pedido do módulo CADP RX; (3) verifica os pacotes de confirmação recebidos pelo CADP RX, reenviando o ultimo pacote quando se verificam inconsistências no pacote de confirmação recebido ou se o pacote de confirmação não for recebido dentro de um limite de tempo pré-estabelecido.

Os sinais ilustrados na Fig. 5 são utilizados para trocar informação entre o NIR e os núcleos. Para enviar informação através da ligação *i*, o NIR utiliza um sinal de pedido *link\_out\_req\_i*, um sinal de confirmação *link\_out\_ack\_i* e uma palavra de dados *link\_out\_word\_i*. A palavra de dados *link\_out\_word\_i* é válida apenas quando o sinal *link\_out\_req\_i* está activo. O núcleo receptor deve gerar o sinal de confirmação *link\_out\_ack\_i* para validar os dados recebidos e permitir a recepção da próxima palavra. Para receber informação através da ligação *i*, o NIR utiliza um sinal de pedido *link\_in\_req\_i*, um sinal de confirmação *link\_in\_ack\_i*, e uma palavra de dados *link\_in\_word\_i*. A palavra de dados é válida apenas quando o sinal de pedido *link\_in\_req\_i* está activo. Como o NIR pode estar ocupado

enviando ou reenviando um pacote anterior, o sinal *link\_in\_ack\_i* é utilizado para confirmar ou não a última palavra recebida. Rajadas de dados são suportadas em ambas as direcções activando o sinal de confirmação de forma contínua, onde o sinal de confirmação num ciclo corresponde ao sinal de pedido recebido no ciclo anterior.

A estrutura preferencial de uma palavra de dados está ilustrada na Fig. 6. Esta é constituída por três campos: endereço, leitura/escrita e dados. O único campo obrigatório é o campo de dados. Endereço e leitura/escrita podem ser omitidos se não forem necessários. Cada ligação tem como destino um conjunto de registos que podem ser acedidos para leitura e escrita. Se o NIR enviar um pedido através da ligação *link\_out\_i* ele irá colocar o endereço de leitura no campo de endereço e definir o sinal de leitura/escrita de acordo com a operação desejada. O núcleo ao receber este pedido, irá responder colocando o valor do registo endereçado no campo de dados do *link\_in\_i*. O tamanho em bits dos campos de endereço não está definido, e depende apenas das necessidades das entidades que estão a trocar esses dados.

Os sinais que ligam um núcleo em particular ao NIR são ilustrados na Fig. 7. Nas palavras que chegam ao núcleo e são enviadas pelo núcleo temos os três campos recomendados: endereço, escrita/leitura, e dados. A ligação de entrada pode escrever ou ler os diferentes endereços. No caso de uma leitura, o núcleo retorna os dados lidos no campo de dados da ligação de saída. O endereço de saída pode ser utilizado, por exemplo, para distinguir as respostas a um pedido recebido de um pedido efectuado pelo núcleo.

Cada núcleo inclui obrigatoriamente um registo com informação de identificação (Core ID Register) que pode ser endereçado para leitura em cada ligação utilizada por esse núcleo. Este registo deve identificar univocamente o núcleo, e a ligação, no caso de o núcleo possuir mais de uma ligação de entrada. Outros registos endereçáveis, recomendados mas não obrigatórios, são os seguintes:

1. Registos de Controlo: utilizados para receber comandos da interface de acesso núcleo.
2. Registos de Estado: utilizados para informar sobre o estado do núcleo
3. Registos de Configuração: utilizados para estabelecer o modo de operação a ser utilizado nas operações seguintes
4. Registos de Dados: utilizados para enviar/receber informação funcional ou de verificação para/de os núcleos
5. Outros registos do utilizador

O registo de identificação do núcleo deve também informar sobre o número de núcleos de propriedade intelectual que se encontram ligados ao próprio, com a finalidade de esses serem potencialmente acedidos. Notar que o esquema de acesso que se está a propor é do tipo hierárquico como mostrado na Fig. 2. No caso em que um núcleo de propriedade intelectual esteja ligado a outros núcleos filhos ou pais, é imperativo que este tenha de conter um outro registo, o Registo de Selecção de Filho, que possa ser escrito de forma a ser seleccionado um núcleo filho específico a que se pretenda aceder. A sequência de passos necessários para aceder a um núcleo filho é ilustrada na Fig. 8. Em primeiro lugar o Registo de Identificação do núcleo pai deverá ser

lido para identificação do número de ligações a núcleos filhos. Em segundo lugar, o número do núcleo filho é escrito no Registo de Selecção de Filho para escolher o Núcleo Filho que se pretende aceder. Em terceiro lugar, a partir desse momento o NIR passa a aceder aos registos no interior do núcleo filho, e não aos do núcleo pai. Em quarto lugar, para voltar a ganhar acesso ao núcleo pai, bastará escrever o número -1 no Registo de Selecção de Filho do próprio núcleo filho. A partir deste momento o NIR comunica novamente com o núcleo pai. A forma como o núcleo filho é acedido pelo NIR não é especificada neste documento, mas poderá ser efectuada através de um comportamento transparente por parte do núcleo pai ou através de registos em série (*pipeline*).

A hierarquia das camadas do protocolo é mostrada na Fig. 9, começando pela camada Ethernet e acabando na camada CADP. A camada física foi excluída para simplificação. No caminho de recepção, as camadas retiram sucessivamente os cabeçalhos e os rodapés dos pacotes, processando a informação contida nos mesmos, até que a informação chegue aos núcleos na forma de palavras de ligação. No caminho de transmissão, a informação é enviada a partir dos núcleos na forma de palavras de ligação que por sua vez são transformada pelas camadas, onde vão sendo adicionados os cabeçalhos e os rodapés respectivos a cada camada.

Os formatos possíveis do pacote CADP são mostrados na Fig. 10. Existem no momento três formatos para esses pacotes: o pacote de palavra única, o pacote de palavras múltiplas e o pacote de envio de comandos. Os campos de um pacote CADP são explicados no texto em baixo:

**Número do Pacote [16 bits]:** todos os pacotes são numerados para evitar a perda de dados usando um mecanismo simples de pedido e confirmação (*handshaking*).

**Tipo do Pacote [16 bits]:** Identifica a função do pacote, que correntemente poderá ser uma das seguintes:

- o *Pacote de palavra única* - contem uma única palavra de ligação de 48 bits.
- o *Pacote de palavras múltiplas* - Contém até 240 palavras de ligação.
- o *Início de Ligação* - inicia o NIR para comunicação, colocando-o em modo de acesso aos núcleos. Este tipo de pacote apenas pode ter como fonte uma estação de trabalho (computador) de um utilizador.
- o *Fim de Ligação* - desactiva o NIR para fins de comunicação, colocando-o em modo normal de operação. Este tipo de pacote apenas pode ter como fonte uma estação de trabalho de um utilizador.

No futuro podem ser adicionados outros tipos de pacotes de forma a implementar novas funções.

**Identificação da Ligação [16 bits]:** identifica a fonte ou o destino da ligação.

**Palavra de ligação [48 bits]:** contem a informação recebida ou entregue à ligação.

O NIR é acedido remotamente a partir de uma estação de trabalho onde o programa que controla o NIR está instalado. O controlador está implementado no topo do protocolo UDP, usando comunicações baseadas em *sockets*.

O pseudo código das funções básicas do programa que controla o NIR é:

```

CADP_send (buffer_out){
//buffer_out is {packet_type, packet_number, link_id,
link_data};
    while      (!(buffer_in.packet_number      ==
buffer_out.packet_number &&
                    buffer_in.packet_type == ack)){
        send (socket, buffer_out);
        while (receive (socket, buffer_in) == 0)
            sleep(TIMEOUT);
    }
    buffer_out .packet_number ++;
}

```

```

CADP_receive (buffer_in){
    while (receive (socket, buffer_in)){
        if( buffer_in.packet_type != ack )
            break;
        sleep(TIMEOUT);
    }
    buffer_out = {ack, buffer_in.packet_number};
    send (socket, buffer_out);
}

```

Este pseudo código descreve as funções *CADP\_send()* e *CADP\_receive()* para enviar/receber pacotes de informação para/de um núcleo particular de um SI, incluindo o próprio NIR. As funções *CADP\_send()* e *CADP\_receive()* usam as funções de baixo nível *send()* e *receive()* do sistema operativo, que utilizam comunicação baseada em *sockets* e

recorrem ao conhecido protocolo UDP. A implementação das camadas ARP, IP e UDP no hardware do NIR é a que permite mais facilmente uma comunicação com ele utilizando operações padrão de chamadas ao sistema numa estação de trabalho remota. Utilizando um protocolo de comunicação de baixo nível é possível reduzir o hardware necessário no NIR. No entanto, isto requer o conhecimento detalhado do controlador da interface de rede da estação de trabalho, o que pode ir para além do que o sistema operativo oferece e tornar-se, conseqüentemente, dependente do sistema. A utilização das funções do sistema operativo garante a independência do sistema.

A função *CADP\_send()* assume que o pacote a ser enviado já se encontra formatado e pronto na estrutura *buffer\_out*. Envia o pacote para fora e espera até receber o pacote de confirmação de recepção desse pacote: verifica se o pacote recebido é do tipo *ack* e se o número corresponde ao número do pacote enviado para fora. Quaisquer outros tipos de pacote recebidos entretanto são descartados, e após *TIMEOUT* segundos tenta reenviar o pacote. Este processo é repetido até receber o pacote de confirmação de recepção do pacote que se está a transmitir.

A função *CADP\_receive()* tenta receber um pacote que não seja um pacote de confirmação de recepção. Se for um pacote de confirmação de recepção este será descartado e após *TIMEOUT* segundos tenta receber de novo um pacote que não seja de confirmação de recepção. Depois receber e tratar um pacote que não seja de confirmação de recepção a função envia um pacote de confirmação de recepção correspondente ao pacote que acabou de receber.

Funções de mais alto nível para controlo do dispositivo NIR podem ser construídas a partir das funções básicas *CADP\_send()* e *CADP\_receive()*. Algumas funções de mais alto nível importantes são:

- *set\_NIR\_mode(mode)*: altera o NIR para o modo de acesso a núcleos ou modo de operação normal conforme o valor do argumento *mode*
- *list\_level1\_cores()*: lista os identificadores (IDs) dos núcleos que se encontram conectados directamente às ligações do NIR
- *list\_children(link\_id)*: lista os identificadores dos núcleos filho que estão ligados ao núcleo que se encontra actualmente conectado à ligação *link\_id* do NIR
- *push\_path(link\_id, child\_link\_id)*: altera o núcleo actualmente conectado à ligação *link\_id* para o núcleo filho conectado ao *child\_link\_id*
- *pop\_path(link\_id)*: altera o núcleo actualmente conectado na ligação *link\_id* para o seu núcleo pai
- *extract\_access\_tree()*: pesquisa todos os nós e ramificações da árvore de acesso e constrói uma representação dessa árvore compreensível pelo utilizador.

No futuro novas funções avançadas podem ser construídas utilizando a biblioteca de funções existente.

## **Referências**

- [1] IEEE Std 1149.1 Standard Test Access Port and Boundary-Scan Architecture

- [2] Philips Semiconductors, "The I<sup>2</sup>C Bus Specification", Version 2.1, January 2000.
- [3] Motorola, SPI bus specifications
- [4] Xilinx, "Architecting Systems for Upgradability with IRL (Internet Reconfigurable Logic)", application note XAPP412, Version 1.0, June, 2001.
- [5] IEEE Std 1500 Standard for Embedded Core Test

Lisboa, 12 de Junho de 2007

### REIVINDICAÇÕES

1. A arquitectura de acesso a núcleos por rede para aceder a núcleos de propriedade intelectual individuais em sistemas integrados, para finalidades funcionais e não funcionais, através de uma rede de comunicação, **caracterizada por** incluir:

- o Núcleo de Interface de Rede que em funcionamento normal é utilizado como um vulgar núcleo de controlo de acesso ao meio ligado a um processador embebido que implementa as camadas mais altas do protocolo de comunicação: protocolo de Internet, protocolo de controlo de transporte, protocolo de datagrama do utilizador, utilizando um vulgar interface de controlo do acesso ao meio e no modo de operação de acesso ao núcleo comunica com núcleos de propriedade intelectual do sistema integrado; tem ainda uma interface para o núcleo ou para um circuito externo responsável pela implementação da camada física do protocolo;
- uma árvore de acesso a núcleos de propriedade intelectual onde os núcleos que estão directamente ligados ao NIR são designados por núcleos de nível 1 e, generalizando, os núcleos ligados a núcleos de nível  $j$  são núcleos de nível  $j+1$ ;
- uma estação de trabalho que consiste numa biblioteca de funções de acesso a núcleos e aplicações em software remotas de núcleos de propriedade intelectual individuais, que acedem remotamente ao software que controla a arquitectura de acesso e é implementada em cima do

protocolo de datagrama do utilizador, utilizando uma comunicação baseada em *sockets*.

2. O núcleo de interface de rede, de acordo com a reivindicação 1, **caracterizada por** o protocolo de pilha implementado no hardware consistir numa camada superior de protocolo projectada para interagir com outros núcleos, à excepção de processadores embebidos, chamado Protocolo de Datagrama de Acesso ao Núcleo, CADP, que consiste num bloco CADP Rx que entrega a informação recebida da rede aos núcleos e num bloco CADP Tx que entrega a informação recebida dos núcleos à rede.
3. A camada superior do protocolo, de acordo com a reivindicação 2, caracterizada por compreender pacotes de informação constituídos pelos seguintes campos: tipo do pacote, número do pacote, identificador de ligação e pelo menos uma palavra da ligação.
4. A árvore de acesso a núcleos, de acordo com a reivindicação 1, **caracterizada por** em cada ramo da árvore consistir numa interface paralela bidireccional, com sinais de pedido e confirmação em cada sentido.
5. A interface paralela de acordo com a reivindicação 4, **caracterizada por** cada palavra ser constituída por campos de endereço, de escrita/leitura e de dados.
6. A arquitectura de acesso, de acordo com a reivindicação 1, **caracterizada por** os núcleos

acessíveis conterem obrigatoriamente um registo de identificação do núcleo e um registo de selecção de filho para fins de acesso à informação dos filhos ou do pai, à excepção do núcleo da interface de rede, na árvore de acesso.

7. A arquitectura de acesso, de acordo com a reivindicação 1, **caracterizada por** ser permitido estabelecer ou desfazer um caminho de comunicação da raiz da árvore do núcleo de interface de rede ao núcleo alvo escrevendo no registo de selecção de filho de cada núcleo no trajecto.
8. Uma arquitectura de software que funciona num terminal de trabalho remoto do utilizador, de acordo com a reivindicação 1, **caracterizada por** permitir que aplicações do utilizador naveguem através da arquitectura de acesso, descobrindo núcleos ligados à árvore de acesso, e permitindo que aplicações específicas a esses núcleos comuniquem com eles.
9. A arquitectura de software, de acordo com a reivindicação 8, **caracterizada por** compreender funções básicas de controlador para enviar e receber pacotes para arquitectura de acesso.
10. A arquitectura de software, de acordo com a reivindicação 9, **caracterizada por** compreender funções de mais alto nível que as funções básicas do controlador, incluindo funções para programar o modo de operação do núcleo de interface de rede; listar núcleos do nível 1; listar os filhos do núcleo alvo;

obter a árvore de acesso completa; ligar a um filho de um núcleo; e retomar a ligação com o núcleo pai.

11. O núcleo de interface de rede, de acordo com a reivindicação 1, **caracterizado por** compreender um protocolo de rede de alta velocidade, incluindo uma rede local Ethernet multi-giga bit.
12. A arquitectura de acesso a núcleos por rede, de acordo com a reivindicação 1, **caracterizado por** os núcleos acessíveis conterem registos de configuração de múltiplo endereçamento, que permitem reconfiguração parcial do núcleo efectuando a reconfiguração acedendo aos núcleos separadamente.
13. A arquitectura de acesso a núcleos por rede, de acordo com a reivindicação 1, **caracterizado por** os núcleos acessíveis serem acedidos usando os seus relógios de sistema internos, não necessitando de nenhum relógio específico para propósitos de configuração, simplificando deste modo a reconfiguração durante a execução, se for necessária.
14. A arquitectura de acesso a núcleos, de acordo com a reivindicação 1, **caracterizado por** os núcleos de propriedade intelectual serem acedidos para configuração, verificação ou propósitos de teste, sem hardware externo para detecção de erros, acessórios de teste, invólucros ou cabos especiais.

Lisboa, 12 de Junho de 2007

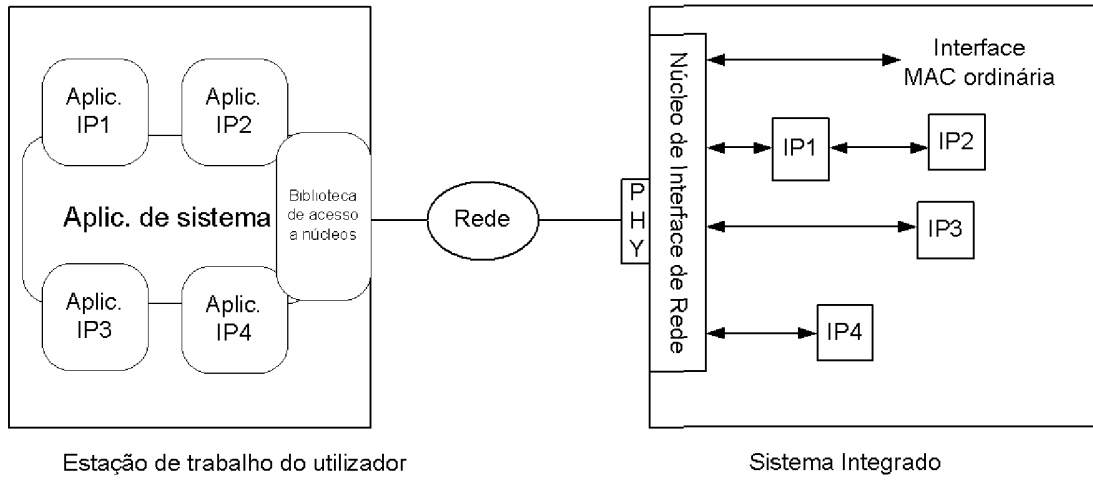


Fig. 1

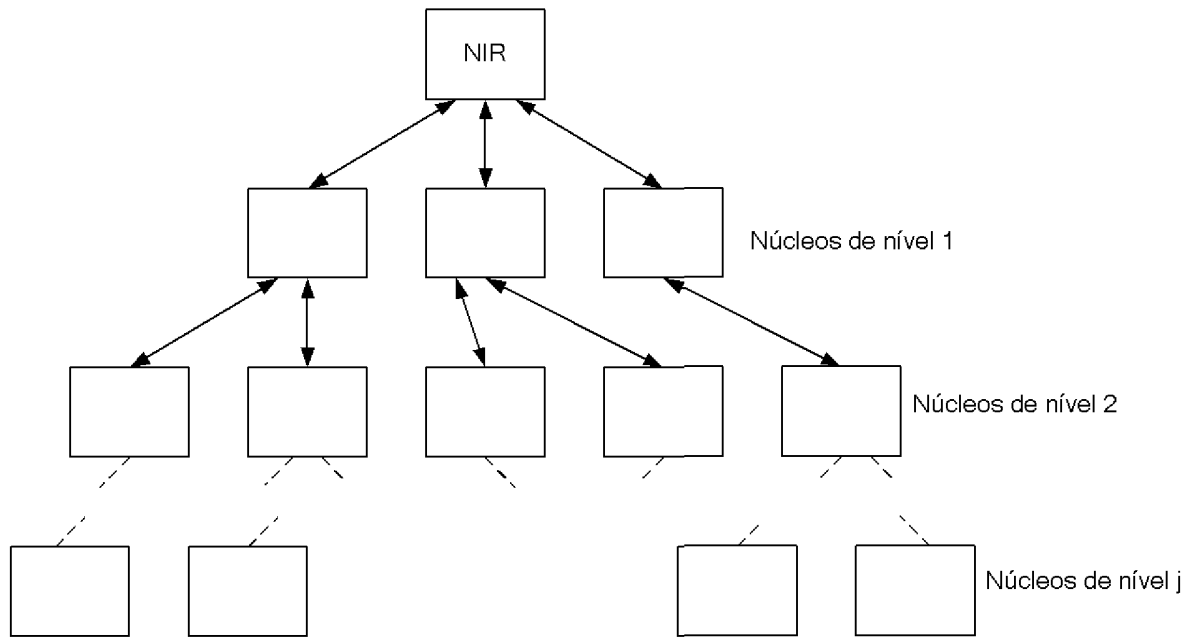


Fig. 2

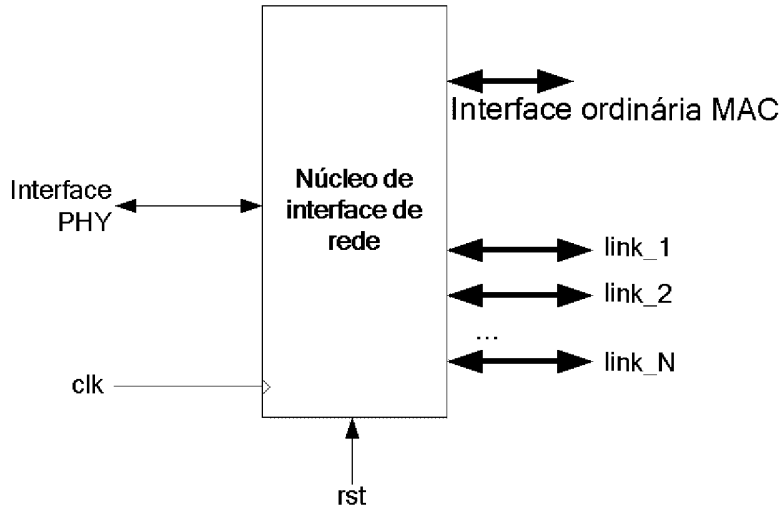


Fig. 3

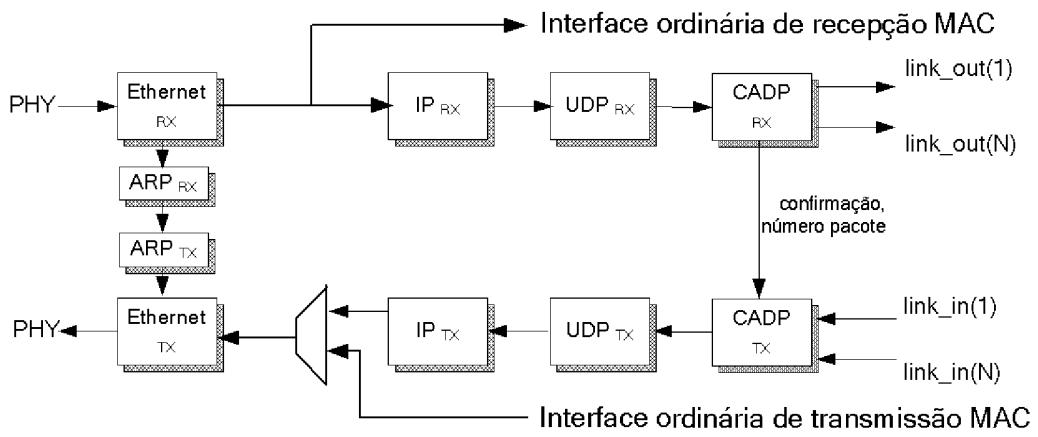


Fig. 4

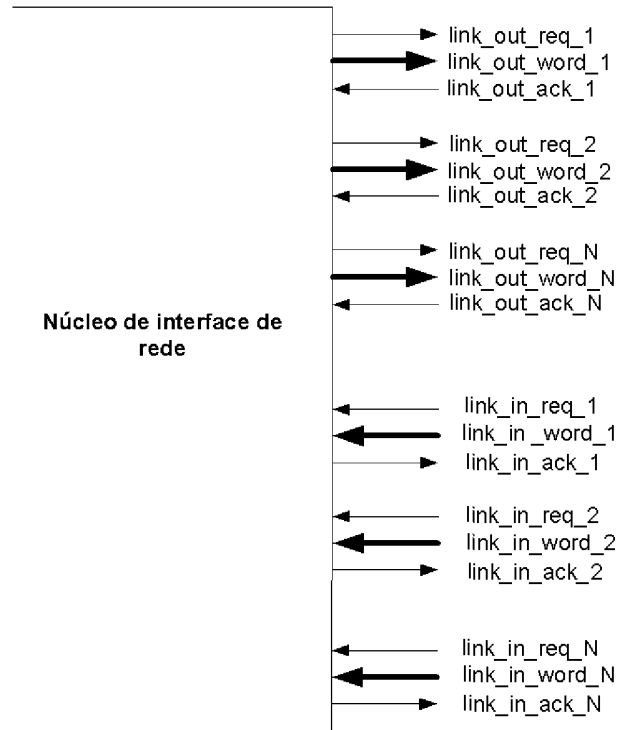


Fig. 5

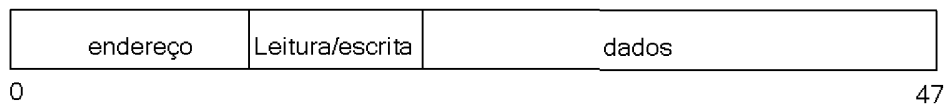


Fig. 6

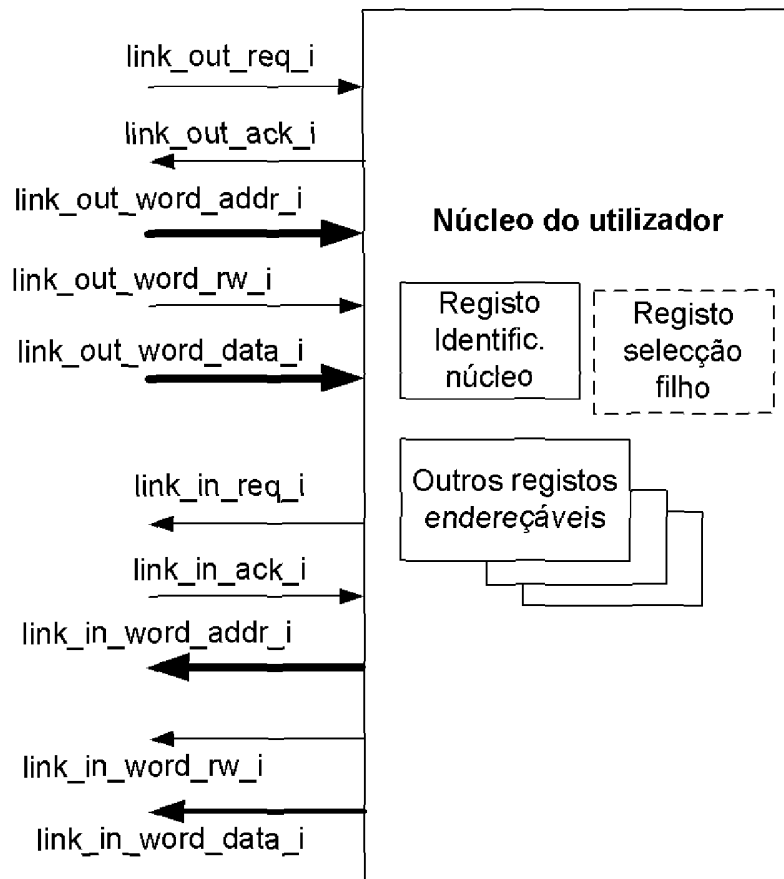
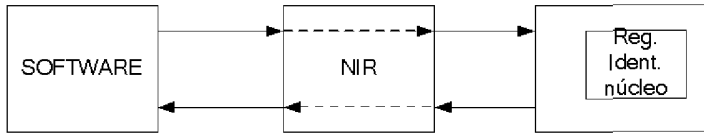
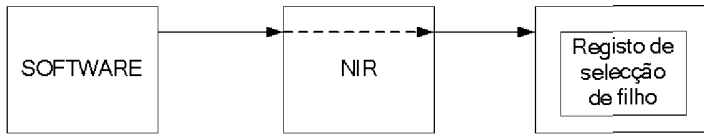


Fig. 7

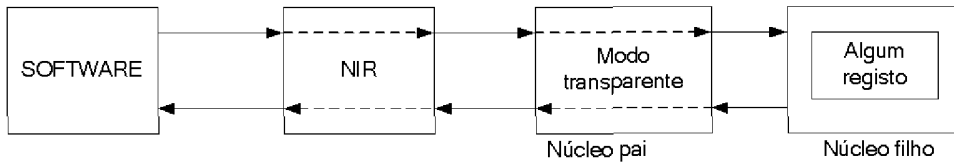
Passo 1: ler o registo de indentific. de núcleo para saber o número de filhos



Passo 2: seleccionar o núcleo filho escrevendo o seu número no registo de selecção de filho



Passo 3: aceder a registos no núcleo filho



Passo 4: devolver o controlo ao núcleo pai escrevendo -1 no registo de selecção de filho do filho

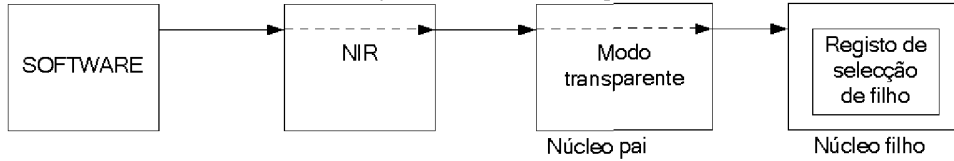


Fig. 8

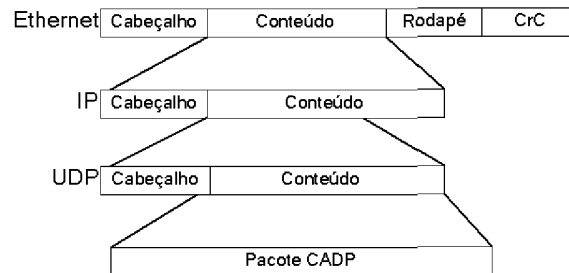


Fig. 9

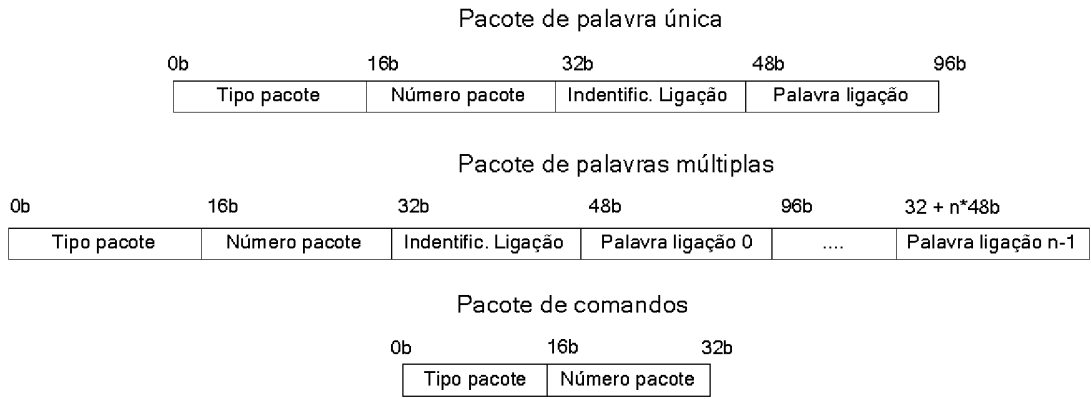


Fig. 10