

Sistema de Teste de Algoritmos de Recuperação de Relógio em Redes ATM 'Uso de Componentes do Tipo EPLD da Altera'

José Manuel Cabral

email: cabral@dei.uminho.pt

Departamento de Electrónica Industrial da Universidade do Minho
INESC - Instituto de Engenharia de Sistemas e Computadores

Resumo

O documento apresenta a descrição de um Sistema desenvolvido para efectuar o teste de Algoritmos de Recuperação de Relógio de Serviços de Débito Binário Constante, em Redes ATM. O Hardware desenvolvido foi essencialmente suportado por dispositivos lógicos programáveis do tipo EPLD (Família 7000) do fabricante Altera. O sistema apresentado constitui um exemplo prático de aplicação destes componentes no desenvolvimento de sistemas constituídos maioritariamente por funções lógicas (e.g. equações booleanas, circuitos sequenciais, máquinas de estado). O sistema de desenvolvimento MAXPLUS II não só permitiu desenhar os blocos funcionais como também simular o seu funcionamento (lógico e temporal). Com efeito, esta ferramenta permite economizar gastos de tempo no *test and debugging* necessários no desenvolvimento destes sistemas num passado bem recente.

A descrição exaustiva do sistema não será feita neste contexto uma vez que o objectivo deste documento é evidenciar as particularidades das ferramentas do sistema e mostrar alguns exemplos de implementação tais como: Máquinas de Estado, Circuitos Combinatórios e Sequenciais (e.g. Contadores, *Shift-Registers*), Circuitos Aritméticos e Multiplexers.

O documento está estruturado em 6 secções. Nas duas primeiras é efectuado o enquadramento do sistema desenvolvido na área das redes de comunicação de banda larga além da descrição funcional do sistema e suas aplicações. Na secção 3 é feita a introdução aos dispositivos lógicos programáveis usados (EPM7096LC-12) e na secção 4 são apresentados alguns exemplos de funções implementadas. A secção 5 apresenta alguns exemplos do uso da ferramenta de simulação comparando os resultados desta com os verificados experimentalmente. Finalmente são apresentadas algumas conclusões, evidenciando as vantagens do uso deste tipo de componentes.

1. Introdução

A evolução para a futura RDIS (RDIS, Rede Digital com Integração de Serviços) de Banda Larga [3] [4] passa pela necessidade desta se poder interligar com as redes já existentes, usando uma tecnologia de transporte baseada em circuitos de Débito Binário Constante (CBR, *Constant Bit Rate*). O transporte de Serviços CBR, na rede ATM, é vulgarmente designado por Emulação de Circuitos.

A Emulação de Circuitos em Redes ATM permite o envio de um sinal de uma rede síncrona através de uma rede assíncrona. Torna-se por isso necessário garantir um mecanismo capaz de sincronizar o fluxo de informação, entre o emissor e o receptor, ou seja, recuperar o Relógio de Emissão no Receptor.

A Recuperação de Relógio é uma "peça" fundamental no conceito da Emulação de Circuitos em Redes ATM. Os Métodos de Recuperação de Relógio requerem um estudo profundo, principalmente quando entre o Emissor e o Receptor não existe uma referência temporal comum. Por outro lado, o desconhecimento de alguns parâmetros estatísticos da rede ATM torna necessário o recurso a modelos de simulação que por si só não são suficientes para garantir o desempenho de um dado método [1].

Com o intuito de contribuir para este estudo, foi desenvolvido um Sistema de Teste de Algoritmos de Recuperação de Relógio que permite realizar os testes de hardware necessários à validação de um dado algoritmo [2].

Grande parte dos blocos funcionais deste sistema foram desenvolvidos à custa de dispositivos lógicos programáveis do tipo EPLD (EPLD, Erasable Programmable Logic Devices), sendo um exemplo prático de aplicação destes componentes no desenvolvimento de circuitos constituídos maioritariamente por blocos digitais.

Assim, este documento, evidenciará principalmente as particularidades desta ferramenta à custa de exemplos de aplicação usados no desenvolvimento deste sistema, sendo a descrição funcional detalhada encontrada em [2].

2. Descrição do Sistema

O sistema é composto pelos módulos necessários à adaptação de um serviço de débito binário constante a 2 048 kbit/s (1ª hierarquia PCM europeia) [15] a uma rede ATM, sendo o módulo de recuperação de relógio (a testar) um módulo substituível na Unidade de Recepção. As Unidades de Inter-funcionamento efectuam a adaptação do formato de dados do serviço à Rede ATM, implementando os protocolos definidos para a Camada AAL-Tipo 1 (AAL, *ATM Adaptation Layer*) [14] do Modelo de Referência de Protocolos da RDIS de Banda Larga. A Unidade de Recepção efectua, entre outras funções, o controlo de erros e a detecção de células perdidas e mal inseridas. O Sistema Simulador de Rede não é mais do que um gerador de atrasos programáveis aplicados às células geradas na Unidade de Emissão, incluindo um Módulo de medição do *Wander* do relógio recuperado [13].

O sistema desenvolvido consiste em duas Unidades de Inter-funcionamento de um serviço de Emulação de Circuitos a 2 048 kbit/s [8], uma de emissão e outra de recepção, como ilustra a Figura 1:

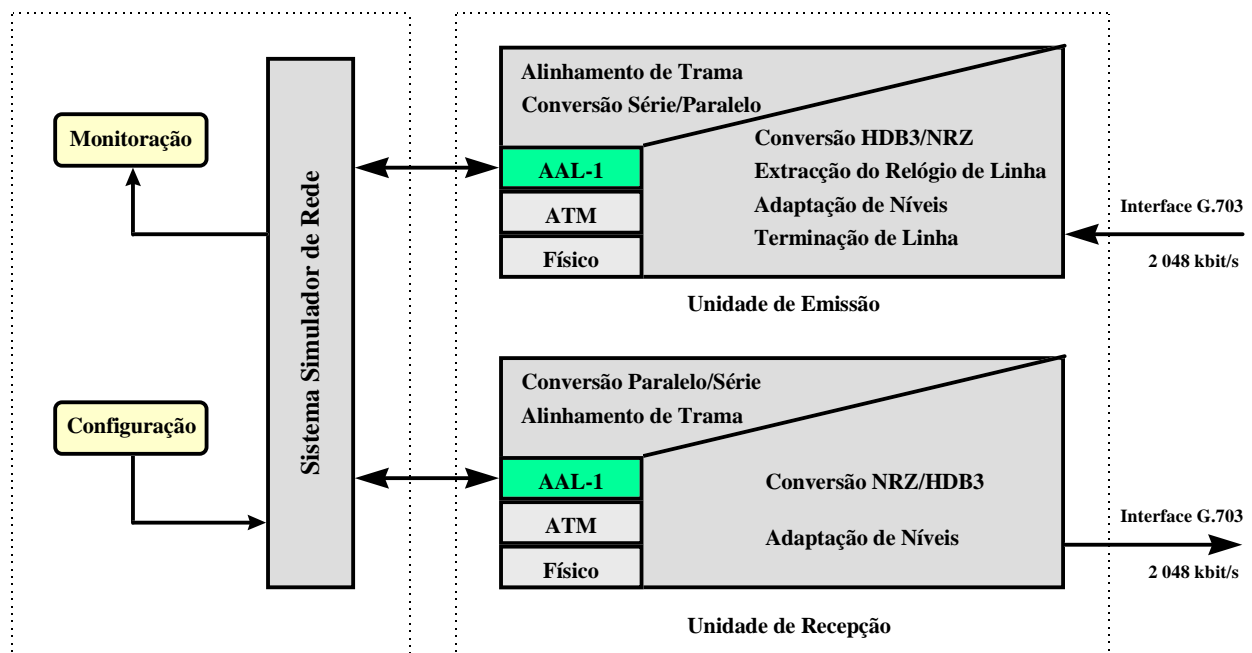


Figura 1: Estrutura do Sistema desenvolvido na perspectiva do Modelo de Referência de Emulação de Circuitos do ATM Forum.

O algoritmo de recuperação de relógio será testado no módulo AAL da Unidade de recepção. A carta a testar receberá todos os sinais necessários para efectuar a recuperação de relógio, sendo a interface com a carta principal efectuada através de *Piggy Back*.

A interface a 2 048 kbit/s (emissão e recepção) segue a recomendação G.703 do ITU [10].

O Sistema de Simulação de Rede além de substituir a rede ATM, permite a configuração de cenários particulares, essenciais para o teste dos algoritmos de recuperação de relógio. Este bloco introduz um atraso programável pelo utilizador aos diversos SAR-PDU's gerados na emissão, permitindo obter um efeito semelhante ao atraso sofrido pelas células numa Rede ATM.

3. Considerações sobre os componentes usados

Nesta secção serão tecidas algumas considerações sobre os componentes usados na implementação dos diversos módulos do sistema desenvolvido.

Em primeiro lugar, salienta-se o facto da grande maioria dos módulos efectuar exclusivamente funções lógicas digitais (operações *booleanas*, contadores, máquinas de estado, etc). Assim, seria conveniente desenvolver estes módulos recorrendo a dispositivos lógicos programáveis com capacidade de integração adequada para implementar as funções necessárias, evitando o recurso às famílias lógicas do tipo TTL, pouco flexíveis e com baixa capacidade de integração.

A existência no mercado de uma grande variedade de soluções de dispositivos de integração lógica, como por exemplo os FPGA's (FPGA, *Field Programmable Gate Array*) da Xilinx e os CPLD's (CPLD, *Complex Programmable Logic Devices*) da Altera, fornecem ao projectista um número elevado de vantagens, destacando-se as seguintes:

Elevado grau de Integração associado a um baixo custo: a existência de dispositivos com grande capacidade de integração lógica permite reduzir o tamanho e a complexidade das cartas e, conseqüentemente, o seu custo global;

Desempenho Elevado: o elevado grau de integração conduz ao uso de um menor número de componentes e de ligações a efectuar, diminuindo o risco de falhas;

Tempos de Desenvolvimento mais curtos: o *software* associado para o desenvolvimento destes sistemas permite reduzir o tempo global de projecto de uma forma substancial. Com efeito, o tempo despendido no desenho esquemático, na compilação, verificação e programação é bastante mais curto comparado com as várias fases de projecto que usam dispositivos lógicos discretos;

Ferramentas de Simulação associadas: o sistema de desenvolvimento destes dispositivos inclui normalmente uma ferramenta de simulação bastante fiável e fácil de utilizar. Enquanto que, com componentes discretos certo tipo de correcções implicam o desenho de uma nova carta, com estes dispositivos basta efectuar a correcção do erro (alterar um ficheiro de dados), compilar e reprogramar.

Os dispositivos lógicos programáveis usados são do fabricante Altera e possuem a designação EPM7096LC-15 [7]. Estes dispositivos pertencem à família MAX 7000 deste fabricante e caracterizam-se por utilizarem tecnologia do tipo CMOS (CMOS, *Complementary Metal Oxide Silicon-gate*) e serem do tipo EEPROM (EEPROM, *Electrically Erasable Programmable Read Only Memory*). Estas EPLD's (EPLD, *Erasable Programmable Logic Device*) emulam as funções existentes nos circuitos TTL e permitem a integração de funções lógicas do tipo SSI (SSI, *Small Scale Integration*), MSI (MSI, *Medium Scale Integration*), LSI (LSI, *Large Scale Integration*) e configurações que suportam no máximo 76 entradas ou 72 saídas.

A linguagem AHDL (AHDL, *Altera Hardware Description Language*) é uma ferramenta poderosa e flexível que permite descrever um bloco através de uma linguagem de programação simples, que suporta funções do tipo: equações *booleanas*, tabelas, máquinas de estado e operações aritméticas. Cada bloco funcional do sistema foi especificado num ficheiro de texto, sendo neste documento apresentados alguns exemplos do uso desta linguagem.

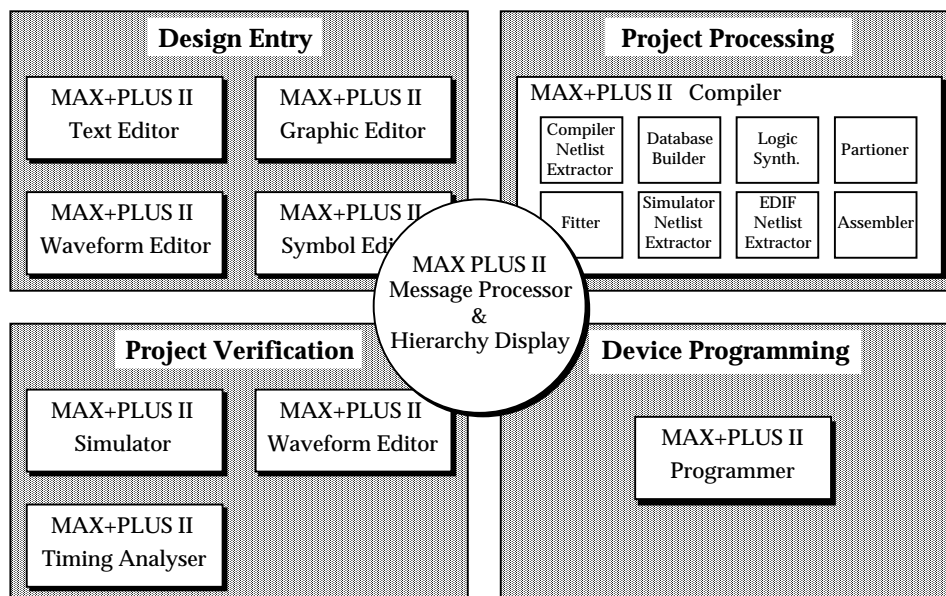


Figura 2: Sistema de Desenvolvimento MAX + PLUS II.

A ferramenta de desenvolvimento - MAX+PLUS II (Figura 2) [5] é constituída por um conjunto de módulos integrados, que permite ao utilizador desenvolver o projecto de um dado circuito digital, de uma forma simples e interactiva [9]. Este sistema existe disponível no mercado em versões Windows para PC e X-Windows para *Workstations* (SUN e HP).

4. Desenvolvimento do Sistema

Nesta secção serão dados alguns exemplos de implementação de blocos do sistema desenvolvido em linguagem AHDL [2]. O critério da escolha destes blocos foi o diversificar o mais possível o número de soluções apresentadas de modo a permitir dar uma perspectiva global do uso desta linguagem.

• Implementação de um Multiplexer de duas entradas e uma saída

A Figura 3 ilustra um exemplo de implementação de um bloco usado para a criação dos SAR-PDU's¹ da Unidade de Inter-funcionamento de Emissão. Esta função é efectuada através de um Multiplexer de duas entradas e uma saída.

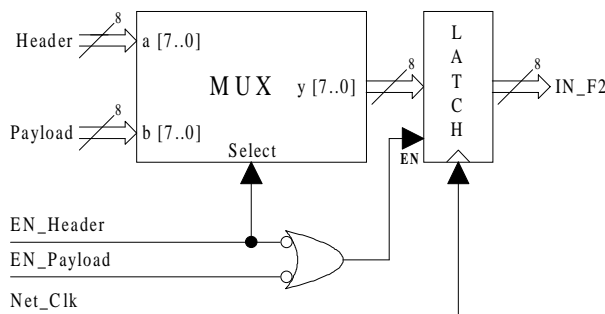


Figura 3: Diagrama funcional do bloco de Criação de SAR-PDU's.

As entradas são dois barramentos de 8 bits que transportam respectivamente, o cabeçalho e o campo de informação (*payload*) do SAR-PDU. A selecção das entradas é efectuada através do sinal EN_Header proveniente de um bloco de Controlo. O barramento de saída do MUX é aplicado a um registo constituído por 8 flip-flops D, accionados pelo sinal de relógio derivado da rede (Net_Clk). A especificação deste bloco em linguagem AHDL é ilustrada na Figura 4.

```
% Criação do SAR-PDU (MUX) %
FUNCTION 2lmux (s, a, b) RETURNS (y);
SUBDESIGN mux
(
    h[7..0], d[7..0], EN_H, EN_P, net_clk    :INPUT;
    In_F2[7..0]                               :OUTPUT;
)
VARIABLE
    flop[7..0]                                :DFFE;
    m[7..0]                                    :2lmux;
    EN_CLK                                     :NODE;
BEGIN
    EN_CLK = EN_H & EN_P;

    m[7..0].a = h[7..0];
    m[7..0].b = d[7..0];
    m[7..0].s = !EN_H;

    flop[7..0].clk = net_clk;
    flop[7..0].ena = !EN_CLK;
    flop[7..0].d = m[7..0].y;

    In_F2[7..0] = flop[7..0];
END;
```

Figura 4: Especificação de um Multiplexer de duas entradas e uma saída.

• Implementação de um detector de nível de enchimento de um FIFO

Este bloco destina-se a determinar o número de SAR-PDU's acumulados no FIFO², ou seja, a diferença entre o número de SAR-PDU's escritos e lidos. Esta operação é efectuada através da detecção dos impulsos activos dos sinais de escrita e de leitura de sequências de 48 bytes (Menos e Mais). A saída deste bloco (LEV[6..0]) indica o número de SAR-PDU's acumulados no FIFO. A sua especificação é apresentada na Figura 5:

¹ SAR-PDU - Segmentation And Reassembly sublayer - Protocol Data Unit

² FIFO - First-In-First-Out Memory

```

% Determinação do Nível do FIFO %
SUBDESIGN fifo_lev
(
  Menos, Mais, Net_Clk           :INPUT;
  Lev[6..0]                       :OUTPUT;
)

VARIABLE
  count[6..0]                     :DFF;
  up, down                         :NODE;

BEGIN
  up = Menos & !Mais;
  down = !Menos & Mais;

  count[0].clk = Net_Clk;

  IF up THEN
    count[0].d = count[0].q + 1;
  ELSIF down THEN
    count[0].d = count[0].q - 1;
  ELSE
    count[0].d = count[0].q;
  END IF;

  Lev[6..0] = count[6..0];
END;

```

Figura 5: Especificação do bloco de Detecção do Nível de enchimento de um FIFO.

A funcionalidade deste bloco é basicamente a de um Contador *Up/Down* de 7 bits.

• Máquina de Estados para Inicialização de FIFO's

Este bloco constitui um exemplo simples de implementação de uma Máquina de Estados criada para efectuar a inicialização de dois FIFO's.

O fabricante especifica que deverá ser aplicado um sinal (Reset) aos FIFO's durante os dois ciclos iniciais de relógio. Deste modo, a máquina percorre os estados s0 a s4 permanecendo no estado s4 até nova inicialização do sistema. O FIFO-1 é inicializado nos estados s1 e s2 e o FIFO-2 nos estados s2 e s3.

```

% Bloco de Inicialização dos FIFO's (RST_FIFO) %
SUBDESIGN RST_FIFO
(
  net_clk                         :INPUT;
  RST_F1, RST_F2                 :OUTPUT;
)

VARIABLE
  start: MACHINE WITH STATES (s0,s1,s2,s3,s4);

BEGIN
  start.clk = net_clk;

  CASE (start) IS
    WHEN s0 =>
      start = s1;
      RST_F1 = VCC;
      RST_F2 = VCC;
    WHEN s1 =>
      start = s2;
      RST_F1 = GND;
      RST_F2 = VCC;
    WHEN s2 =>
      start = s3;
      RST_F1 = GND;
      RST_F2 = GND;
    WHEN s3 =>
      start = s4;
      RST_F1 = VCC;
      RST_F2 = GND;
    WHEN s4 =>
      start = s4;
      RST_F1 = VCC;
      RST_F2 = VCC;
  END CASE;
END;

```

Figura 6: Especificação da Máquina de Estados de inicialização dos FIFO's.

5. Simulação dos Blocos Funcionais do Sistema

Nesta secção é ilustrado um exemplo de simulação de um bloco apresentado na secção anterior - Detector do Nível de Enchimento de um FIFO.

A Figura 7 ilustra o diagrama funcional para melhor compreensão do seu funcionamento:



Figura 7: Diagrama funcional do bloco detector do nível de um FIFO's.

O detector é accionado pelo relógio derivado da rede (Net_clk), sendo a saída (Level[6..0]) função dos sinais “Mais” e “Menos”. Quando o valor do sinal “Mais” é “0” então o valor de Level[6..0] é incrementado de uma unidade, acontecendo o inverso para o sinal “Menos”.

Na Figura 8 está ilustrado o ecrã do editor de formas de onda após se ter efectuado uma simulação com o cenário indicado:

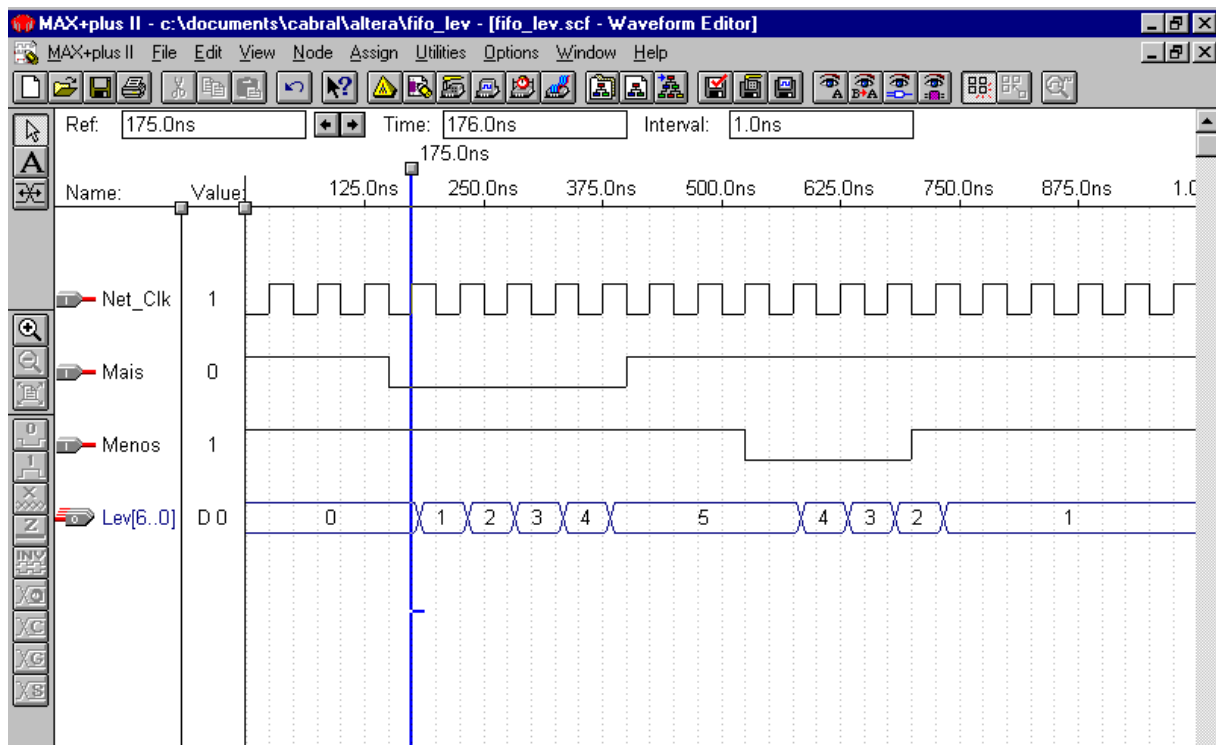


Figura 8: Simulação lógica e temporal do bloco especificado na Figura 5.

Esta ferramenta permite não só a verificação lógica do funcionamento de um circuito digital como também a simulação temporal. Além disso, são verificados outros parâmetros tais como *setup-time* e *hold-time* de registos.

6. Conclusões

A Emulação de Circuitos é um conceito fundamental na evolução das actuais redes para a Rede Digital com Integração de Serviços de Banda Larga. Com efeito, numa fase intermédia as redes actuais vão continuar a existir, sendo necessário garantir que a rede ATM transporte, de uma forma transparente, o tráfego destas redes.

A Emulação de Circuitos consiste no transporte de sinais de uma rede síncrona através de uma rede assíncrona, sendo necessário por isso efectuar a Recuperação do sinal de Relógio de Emissão, no Receptor.

O Sistema de Teste de Algoritmos de Recuperação de Relógio desenvolvido constitui uma ferramenta importante para o teste dos métodos de Recuperação de Relógio na ausência de uma rede ATM. Por outro lado, o teste de algoritmos usando uma rede ATM poderá não introduzir condições extremas necessárias à sua validação.

O objectivo deste documento foi, contudo, evidenciar as particularidades da ferramenta de desenvolvimento de grande parte dos componentes deste sistema, ou seja, o sistema MAXPLUS II da Altera e a sua linguagem de especificação AHDL.

Sem fazer uma descrição exaustiva de toda especificação do sistema, foram dados alguns exemplos de implementação e simulação de blocos funcionais baseados em funções lógicas.

A experiência adquirida ao longo deste trabalho aconselha o uso deste tipo de ferramentas, não só no projecto e desenvolvimento de sistemas deste tipo, como um meio didáctico para o estudo de sistemas digitais em geral.

7. Bibliografia

- [1] Almeida, N.; Cabral, J.; Alves, A.; "End-to-End Synchronization in Packet Switched Networks"; 2nd International Workshop on Network and Operating System Support for Digital Audio and Video"; Heidelberg; Novembro 1991.
- [2] Cabral, J. M.; "Emulação de Circuitos em Redes ATM (Sistema de Teste de Algoritmos de Recuperação de Relógio)"; Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto; Junho 1995.
- [3] De Prycker, Martin, "Asynchronous Transfer Mode - Solution for Broadband ISDN", Ellis Horwood, 1991.
- [4] McDysan, D. E.; Spohn, D. L., "ATM Theory and Application", McGraw-Hill Series on Computer Communications, 1995.
- [5] MAX+PLUS Development System, Altera Corporation, Abril de 1990.
- [6] Altera Applications Handbook, Altera Corporation, Abril de 1992.
- [7] Altera Data Book, Altera Corporation, Agosto de 1993.
- [8] ATM Forum, "ATM Circuit Emulation Services - Version 1.0", Dezembro de 1994.
- [9] Cardoso, P.M., "Lógica Programável - Aplicação em Circuitos de Telecomunicações", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, Janeiro de 1992.
- [10] ITU-T, Rec. G.703 - "Physical/Electrical characteristics of hierarchical digital equipment", Genebra, 1991.
- [11] ITU-T, Rec. I.121 - "Broadband Aspects of ISDN", Genebra, 1991.
- [12] ITU-T, Rec. I.327 - "BISDN Network Functional Architecture", Genebra, 1991.
- [13] ITU-T, Rec. G.823 - "The control of Jitter and Wander within digital networks which are based on the 2048 kbit/s hierarchy", Helsínquia, Março de 1993.
- [14] ITU-T, Rec. I.363 - "BISDN ATM Adaptation Layer (AAL) Specification", Helsínquia, Março de 1993.
- [15] Owen, F.F.E., "PCM and Digital Transmission Systems", Texas Instruments Electronics Series, McGraw-Hill Book Company, 1982.



José Manuel Cabral nasceu no Porto em 1964. Licenciou-se em Engenharia Electrotécnica e de Computadores na Faculdade de Engenharia da Universidade do Porto em 1988. Concluiu o Mestrado em Engenharia Electrotécnica e de Computadores - Ramo de Telecomunicações em 1995. Após a licenciatura desenvolveu actividade de investigação e desenvolvimento no INESC - Porto na área de Redes Integradas de Comunicação. Actualmente exerce as funções de Assistente no Departamento de Electrónica Industrial da Universidade do Minho, em que as actuais áreas de interesse incluem as Redes Industriais e a tecnologia ATM.