Vol. 36 No. 7　　情　報　処　理　　July 1995

解　説

ハードウェア/ソフトウェア・コデザイン

4. 信号処理分野におけるコデザイン†

宮　崎　敏　明††

1. はじめに

信号処理を実時間で行うLSIは，DSP(Digital Signal Processor) とよばれ，80年代前半から盛んに研究開発され，現在ではMPU, ASIC (Application Specific IC) と並んで一つのLSI文化を築いている．

DSPは，直交変換やフィルタリングなど繰返しの多い処理を高速に行う必要があり，その設計にあたっては処理の時間並列性・空間並列性が考慮される．また，初期仕様がハードウェアを抽象化した動作モデルではなく純粋に処理アルゴリズムとして与えられる点に大きな特徴がある．そのため，MPUやASICに対する設計手法とは異なる手法が試みられている．特に，上位レベル合成 (High-level synthesis) や機能合成 (Behavioral synthesis) とよばれるレジスタ転送レベル(Register Transfer Level: RTL) よりも抽象度の高い仕様からハードウェアを合成するCAD (Computer Aided Design) 技術は，DSPをターゲットアーキテクチャにしているものが多い．それは入力仕様を，ハードウェアを抽象化したモデルではなく，実際に実行したい処理アルゴリズムに求めたことによるところが大きい．

本稿では，DSP設計の特徴およびそれを支援するために提案されているいくつかのCAD技術を紹介しつつ，信号処理分野におけるハードウェア/ソフトウェア・コデザイン（以後，単にコデザインとよぶ）の現状および今後の課題について述べる．

2. DSP設計の特徴

2.1 データパス重視

図-1にNTTで開発された音声処理用DSPであるDSSP 1のブロック図を示す[1]．本例でも分かるとおりDSPは，高速処理を実現するために信号処理アルゴリズムで多用される積和演算を専用乗算器（FMPL）とそれと直列に配置されたALU（FALU）を用いて行うことができるようにデータパス構成に工夫が施されている．また，データのアクセス時間を短くするために，データ系バス（DBUS）と命令系バス（PBUS）を別に持つハーバードアーキテクチャ（Harvard Architecture)[2] を採用している．さらに，マルチポートRAMを使用し，アドレッシングのために専用演算回路（AAU）やループカウンタ（LP）を備えているものも多い[3),7)]．

DSPには，実時間処理が求められ，ある時間内に一定量の処理を行わなければならない．そのため，処理の規則性を見い出し時間展開することによって処理をパイプライン化する技法やデータ依存関係のない演算は同時に行うことができるように演算器を複数個持ったデータパス構成をとることもある．DSPの各命令は，複数演算器やメモリ等を同時に制御するために細かなフィールドに分かれた水平形マイクロ構成[3]をとる．よって，制御対象が多い場合は，VLIW（Very Long Instruction Word）プロセッサ[4]の命令とよく似た形となる．

2.2 実データによる評価重視

DSPを設計する際には，処理アルゴリズムの演算精度や演算量を見積もる必要があり，これを誤ると処理品質の劣化や回路規模の増大を招く．そのため，演算精度や演算量の評価は理論的な検討と共に，実データあるいはベンチマークデータ

† Hardware/Software Co-design for Digital Signal Processing by Toshiaki MIYAZAKI (NTT Optical Network Systems Laboratories).

†† NTT光ネットワークシステム研究所

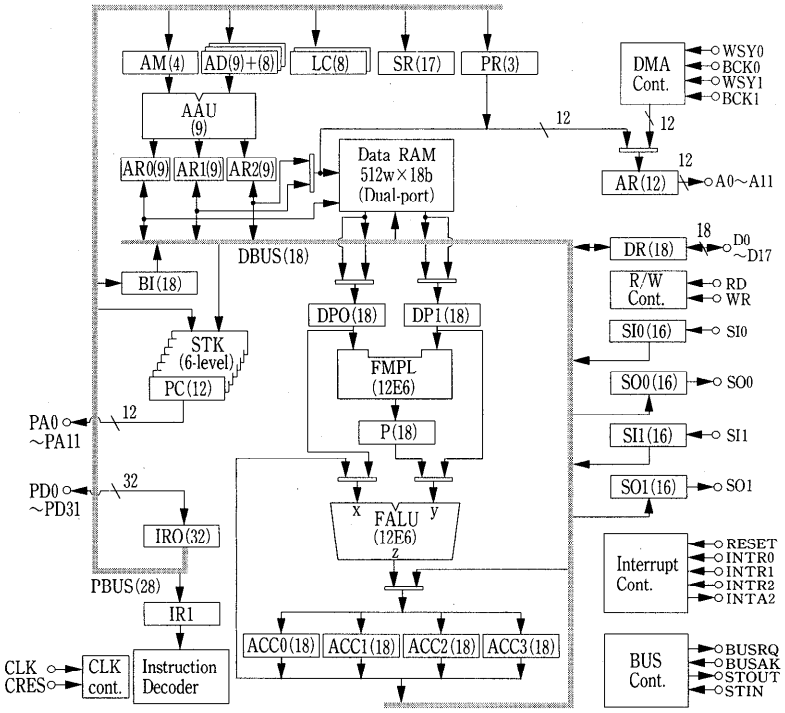


図-1　DSSP 1のブロック図

等を用いてシミュレーションを行うことが多い．

3. コデザインのためのCAD

図-2にDSP設計に関するコデザイン用CADを形成する上で必要と思われる要素技術を示す．各要素技術は今日まで盛んに研究されてきている．しかし，図示したような形でそれらがすべて有機的に統合された実用的なCADシステムは残念ながら現存しない．本章では，それぞれの要素技術を紹介し，次章で統合化への課題を述べる．

3.1 評価・分割技術

性能を重視するDSPの設計支援においては，評価・分割技術が特に重要である．自動分割に関する研究はまだ模索段階であるため，ここでは評価技術を中心に述べる．

仕様定義

ここでは，DSP設計を対象とした仕様定義法としてまずSilage[5),6)]を紹介する．Silageは，信号処理アルゴリズムを効率よく記述するための言語で，カリフォルニア大学バークレー校で開発された．その文法はPascalに似ており，シグナルフローグラフ（SFG）[7)]を自然に書き表せるようになっている．現在，IMECが開発したCATHE-

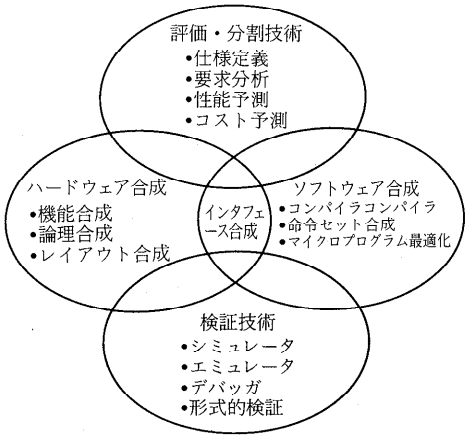
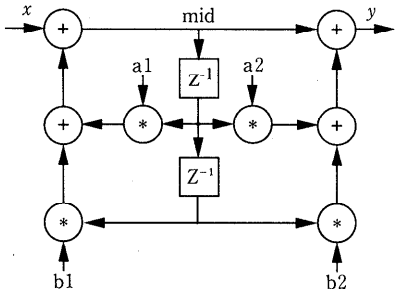


図-2　ハードウェア/ソフトウェア・コデザインを形成する要素技術

DRALシステム[8)~10)]等の入力言語として採用されている．図-3にSilageの記述例を示す．一次遅れを表す@オペレータを備え，ディジタルフィルタ等の記述が簡単に行える．

一方，信号処理アルゴリズム自体の評価は，汎用計算機上で高級言語を用いて行うことが多い．そこで，その高級言語によるプログラムをそのままあるいは少しの情報追加でDSPの仕様として




```
function IIR (a1, a2, b1, b2, x : num)
y : num = /* Return value */
begin
    y=mid+a2*mid@1+b2*mid@2;
    mid-x+a1*mid@1+b1*mid@2;
end
```

図-3 Silage の記述例

使用できないかという考えから，Fortran, Pascal, C 等を入力仕様とする機能合成システムの研究も行われている[11)~13)]．

語長評価

現在，32 bit あるいはそれ以上の語長で浮動小数点演算が可能な DSP も出現しているが，DSP の設計にあたっては本来，系の安定性やスペクトル歪みなどを考慮して各演算をどの程度の語長で行うべきか定める必要がある．演算語長は，理論的に上限を押さえることができる場合もあるが，その場合でも入力信号の偏りから計算で出した演算語長よりも短くてすむ場合もあり，実データを用いて演算語長を見積もることが実用上重要である．上記の目的で行うシミュレーションを，ここでは有限語長シミュレーション（Finite Word Length Simulation: FWLS）とよぶ．FWLS の例として文献 14)，15)がある．FWLS が扱うデータは実際の画像や音声でありその量は膨大である．したがって，その実装にあたっては実行速度を十分考慮する必要がある．

演算量評価

演算精度と並んで演算量を調べることは，対象の処理アルゴリズムを実行する上で，どのような演算が主であり，そのための専用演算器を設けるべきか否かを判断する定量的な情報を得るという意味で重要である．演算量評価には，一般のソフトウェア開発におけるカバレージ・テストと同様に各演算のダイナミックな実行回数をカウントする機構を備えればよい[14)]．

3.2 ハードウェア合成

機能モジュール割当て

既存の機能合成システムは，入力仕様をもとに必要となるすべての機能を合成してしまうものが多い．しかし，DSP 設計を考えた場合，乗算器等キーとなる機能モジュールは人手で詳細設計されたものを使用することが多く，あらかじめ使用可能な機能モジュールを用いる手法の方が現実的である．上記の思想に近いアプローチを行っているシステムとして CATHEDRAL システム[8)~10)]および HYPER システム[16)]があげられる．いずれのシステムも別途用意されたモジュールライブラリに対して入力仕様内の機能を割り当てる処理を含んでいる．

インタラクティブ手法

前述した機能モジュール割当てとも関連するが，完全なトップダウン機能合成ではなく，ユーザが合成の前や途中で積極的にアーキテクチャ決定に関与することによってより良いハードウェア構成を導ける可能性がある．DSP 設計にとって特に重要であるデータパス構成の決定にユーザが積極的に関与できるシステムも研究されている[17),18)]．これらは，DSP のコデザインを考えるとき一つの示唆を与える．

3.3 ソフトウェア合成

DSP は，広義にはフィルタリング等ある信号処理に特化した LSI を含めるが，一般には MPU と同様に命令セットを備えておりマイクロプログラムによって処理内容を変更できる．そのため，高位レベルの入力仕様からマイクロプログラムを生成する開発環境の整備は，新たな DSP を設計するときに並行して考えねばならない重要な事項である．入力仕様として C 言語のような高級言語の他，SFG や伝達関数を単位とするブロック図による入力形態が考えられる．また，命令セットは各命令間の直交性やパイプライン処理を行う際の命令順序を考慮して決定される．マイクロプログラムの自動生成や命令セットを自動決定する試みが行われている[19)~23)]．

3.4 検証技術

検証技術は，3.1 の評価・分割技術を支える技術であり，実用性からシミュレーション・エミュレーション技術の研究が盛んである．

システムシミュレーション

例としてPtolemy[24]システムを紹介する．Ptolemyは，DSPだけでなくA/D, D/A等のアナログ回路を含むヘテロな動作モデルを一括してシミュレーションする機能を提供する．各コンポーネントの動作は，C言語等で記述するが異なる動作モデル間の情報交換を可能にするためにオブジェクト指向に基づいたエンキャプスレーション機能を有する．シミュレーション機能の実装にはUNIXのプロセス間通信機能を用いている．また，SDFというフローグラフ形式の入力仕様からDSPのマイクロプログラムやC言語記述を生成することもできる．さらに，図-2で示したコデザイン環境を目指し，ハードウェア実現部に対してはSilage[5),6)]記述を生成し，HYPER[16)]システムによって自動合成を行う試みもなされている．ただし，現状のシステムでは機能分割をすべて人手で行う必要がある．現在，Ptolemyと類似するツールがいくつか市販されている[25)]．

システムエミュレーション

DSP設計に際しては実データを用いたハードウェアのデバッグおよびソフトウェアの先行開発を目的にモックアップ(Mockup)と称し，汎用個別部品を使用して設計対象のLSIと同一機能を持つ装置を並行して開発することが従来よく行われた．現在では，その用途にICE (In Circuit Emulator)が用いられるが，そのキーとなっているのが80年代に登場したFPGA (Field-Programmable Gate Array) とよばれるプログラマブルデバイスである[26),27)]．FPGAを複数個用いて，ICEを実現することは今日よく行われており，市販品も出回るようになってきた[28),29)]．Intel社が，自社のMPU Pentiumを開発する際に，その一つであるQuickturn Design Systems社のRPM (Rapid Prototyping Machine) を使用したことは有名である．

FPGAを用いたICEの実行速度は，ソフトウェアシミュレータに比べれば桁違いに速いが最終LSIの速度には及ばず何も工夫しないとクロック周波数で10 MHz程度が限度である．ただし，絶対遅延はさほど問題ではなくデータスループットが重要である通信処理回路等の用途では，ラッチ挿入[30)]やリタイミング[31)]等の技法が適用でき，実速度に近い性能を実現することも可能となっている[32)]．

4. 今後の課題

前章で述べた個別CAD技術をコデザインのために統合するには，各CADツールが扱うデータの共有化が図られる必要がある．そのためには，ハードウェアの動作とソフトウェアの動作を統一的に扱うことができる動作モデルおよびその記述法の確立が急務である．

4.1 モデリング

専用ハードウェアとDSPのマイクロプログラムによって構成される信号処理システムの仕様定義は，個別要素の動作仕様とそれらの総体であるシステムの動作仕様とに階層的に分けて行うのがよいと筆者は考える．それは，前者の仕様定義には従来からのハードウェア記述言語 (HDL) やC言語のような手続き型言語を用いるのが設計者にとっては自然であるからである．ただし，システム全体を規定するために個々の要素モデルの違いを吸収する新たな仕様定義方法が必要となる．すなわち先に紹介したPtolemy[24)]が行っているようにそれぞれの要素の定義の上に立つモデルを構築する必要が出てくる．そのモデルの候補として，CSP[33)]等のプロセス代数が考えられる．ただし，信号処理の性質を考えると時間概念を扱えるモデルでなくてはならない．

4.2 プログラマブルハードウェア/CAD/ソフトウェア・コデザイン

信号処理の中にはハフマン符号化のように可変長データを扱わなければならず，本質的に定語長処理を基本とするDSPやMPUでは効率的な処理が望めないものも含まれる．それらの処理への対応として，FPGAが，エミュレータへの応用に留まらず，コプロセッサやアクセレレータのキーコンポーネントとして使用されはじめている[34),35)]．FPGAのようなプログラマブルハードウェア (PHW) を一部に含むシステムを構築するには，**図-4**に示すように，従来のコデザインに加えPHWのためのCADを加えた設計環境を構築する必要がある．PHWはプログラミングされることによって初めて所望の機能を実現することができる．そのため，論理のマッピング・配置・配線等の処理を実行するCADツールを抜きにしては考えることはできず，それらの善し悪しが最

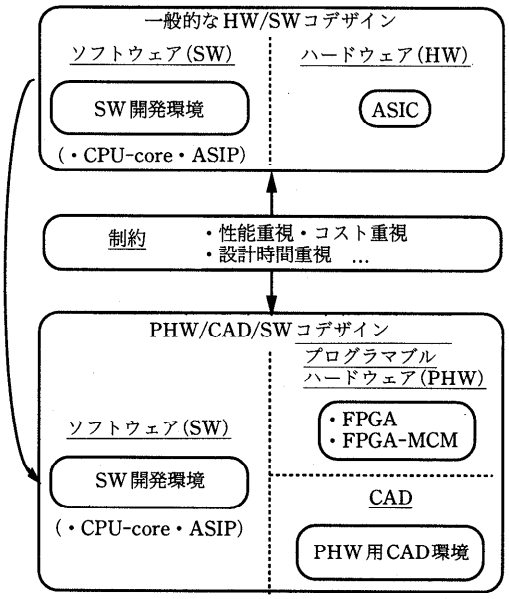


図-4 プログラマブルハードウェア/CAD/ソフトウェア・コデザインの概念図

終的なシステムの性能を支配すると言っても過言ではない．したがって，性能・コスト・設計時間といった様々な制約のもとでバランスのとれたシステムを設計するための環境として図-4に示した設計環境の実現が望まれる．

4.3 コシンセシス

自動設計の立場からコデザインをとらえると，その最終目標は，入力仕様からハードウェア(PHWを含む)およびソフトウェアの分割をある制約に基づいて自動的に行い，さらにそれぞれの部分を自動合成するシステムの実現である．これはコシンセシス(Co-synthesis)ともよばれる[36]．コシンセシス・システムを実現するには，図-2に示したように「評価・分割技術」「ソフトウェア合成技術」「ハードウェア合成技術」およびハードウェアとソフトウェアの同期機構などを含む「インタフェース合成技術」を統合しなければならず，**4.1**で述べたコデザインのためのモデル構築がその鍵を握っている．

5. おわりに

DSPアーキテクチャは，年々複雑さを増している．例えばTI社のDSP TMS 320 C 80は，チップ内部にRISCコアと4つのDSPコアを持ち内蔵共有メモリとクロスバーによって相互接続されたMIMDアーキテクチャとなっている[37]．このような大規模DSPおよびそれを含むシステムを開発していくとき，今までのように個別CADと人手に頼って行う手法では破局を迎えるのは必至である．一人でも多くの読者が信号処理分野におけるコデザインの必要性を認識され，優れたコデザイン・システムが登場してくることを期待する．

参考文献

1) Yamauchi, H., Kaneko, T., Kobayashi, T., Iwata, A. and Ono, S.: An 18-bit Floating-Point Signal Processor VLSI with An on-Chip 512 W dual-Port RAM, Proc. ICASSP'85, pp. 204-207 (1985).
2) Patterson, D. A. and Hennessy, J. L.: Computer Architecture—A Quantitative Approach, Morgan Kaufmann Publishers (1990).
3) 電子情報通信学会編：ディジタル信号処理ハンドブック，オーム社 (1993).
4) Fisher, J. A.: Very Long Instruction Word Architectures and the ELI-512, Proc. IEEE 10th International Symposium on Computer Architecture, pp. 140-150 (June 1983).
5) Hilfinger, P.: A High-Level Language and Silicon Compiler for DSP, Proc. CICC'85, pp. 213-216 (1985).
6) Genin, D., Hilfinger, P., Rabaey, J., Scheers, C. and De Man, H.: DSP Specification Using the Silage Language, Proc. ICASSP'90, pp. 1057-1060 (1990).
7) 青山友紀，小野定康：信号処理プロセッサ，オーム社 (1990).
8) Lanneer, D., Note, S., Depuydt, F., Pauwels, M., Catthoor, F., Goossens, G. and Dc Man H.: Architectural Synthesis for Medium and High Throughput Signal Processing with the New Cathedral Environment, in Camposano, R. and Wolf, W. ed.: High-Level VLSI Synthesis, Kluwer Academic Publishers, pp. 27-54 (1991).
9) De Man, H., Rabary, J., Six, P. and Claesen, L.: CATHEDRAL-II: A Silicon Compiler for Digital Processing, IEEE Design & Test of Computers, Vol. 3, No. 6 (Dec. 1986).
10) Note, S., Chattoor, F., Goossens, G. and De man, H.: Combined Hardware Selection and Pipelining in High-Level Performance Data-Path Design, IEEE Trans. on CAD, Vol. 11, No. 4, pp. 413-423 (Apr. 1992).
11) Tanaka, T., Kobayashi, T. and Karatsu, O.: HARP: Fortran to Silicon, IEEE Trans. on CAD, Vol. 8, No. 6, pp. 649-660 (June 1989).
12) 池永 剛，白井克彦：高級言語により記述されたアルゴリズムを実現する専用プロセッサ設計支援システム，情報処理学会論文誌，Vol. 32, No. 11, pp. 1445-1456 (Nov. 1991).
13) Wakabayashi, K.: Cyber: High Level Syn-

thesis System from Software into ASIC, in Camposano, R. and Wolf, W. ed.: High-Level VLSI Synthesis, Kuluwer Academic Publishers, pp. 127-151 (1991).
14) 田中敏明，小林 勉：信号処理LSI開発用有限語長シミュレータ，信学技報，CAS 85-165 (1986).
15) 雨坪孝尚，上田 穣，吉田 裕，白井克彦：専用プロセッサ支援システム (SYARDS) における大規模システム処理系の設計，情報処理学会全国大会 9M-1, 6-125 p. (Mar. 1993).
16) Chu, C-M., Potkonjak, M., Thaler, M. and Rabaey, J.: HYPER: An Interactive Synthesis Environment for High Performance Real Time Applications, Proc. IEEE International Conference on Computer Design (ICCD'89), pp. 432-435 (Oct. 1989).
17) Knapp, D.W.: Manual Rescheduling and Incremental Repair of Register-Level Detapaths, Proc. IEEE International Conference on Computer-Aided Design (ICCAD'89), pp. 58-61 (1989).
18) Miyazaki, T. and Ikeda, M.: High-Level Synthesis Using Given Datapath Information, IEICE Trans. Fundamentals, Vol. E76-A, No. 10, pp. 1617-1625 (Oct. 1993).
19) Cheng, W-K. and Lin, Y-L.: Code Generation for a DSP Processor, Proc. of the Seventh International Symposium on High-Level Synthesis, pp. 82-87 (May 1994).
20) Schoofs, K., Goossens, G. and De Man, H.: Bit-Alignment for Retargetable Code Generators, Proc. of the Seventh International Symposium on High-Level Synthesis, pp. 76-81 (May 1994).
21) Wilson, T., Grewal, G., Halley, B. and Banerji, D.: An Integrated Approach to Retargetable Code Generation, Proc. of the Seventh International Symposium on High-Level Synthesis, pp. 70-75 (May 1994).
22) Lee, E. A.: Programmable DSP Architectures: Part II, IEEE ASSP MAGAZINE, pp. 4-14 (Jan. 1989).
23) Mahmood, M., Mavaddat, F. and Elmasry, M. I.: Experiments with an Efficient Heuristic Algorithm for Local Microcode Generation, Proc. IEEE International Conference on Computer Design (ICCD'90), pp. 319-323 (Sep. 1990).
24) Kalavade, A. and Lee, E. A.: A Hardware-Software Codesign Methodology for DSP Applications, IEEE Design & Test of Computers, Vol. 10, No. 3, pp. 16-28 (Sep. 1993).
25) Maliniak, L.: EDA Tools Zero in on DSP Design, Electronic Design, Vol. 42, No. 5, pp. 51-60 (1994).
26) Brown, S. D., Francis, R. J., Rose, J. and Vranesic, Z. G.: Field-Programmable Gate Arrays, Kluwer Academic Publishers, Reading (1992).
27) 特集 FPGA—その現状，将来とインパクト，情報処理，Vol. 35, No. 6, pp. 504-540 (June 1994).
28) FPCB Development System Version 1.2 User's Manual, Aptix Corporation, San Jose (1993).
29) Paradigm RP Manual, Zycad Corporation.
30) Miyazaki, T., Nakada, H., Tsutsui, A., Yamada, K. and Ohta, N.: A Speed-Up Technique for Synchronous Circuits Realized as LUT-Based FPGAs, Proc. Fourth International Workshop on Field Programmable Logic and Applications (FPL'94), Prague (Sep. 1994).
31) Leiserson, C. E. and Saxe, J. B.: Retiming Synchronous Circuitry, Algorithmica, Vol. 6, pp. 5-35, Springer-Verlag New York Inc. (1991).
32) Yamada, K., Nakada, H., Tsutsui, A. and Ohta, N.: High-Speed Emulation of Communication Circuits on a Multiple-FPGA System, Proc. ACM Second International Workshop on Field Programmable Gate Arrays (FPGA'94), Berkeley CA (Feb. 1994).
33) Hoare, C. A. R.: Communicating Sequential Processes, Prentice-Hall International (1985).
34) Moore, W., Wayne, L. ed.: More FPGA, Abingdon EE&CS Books, Oxford England (1994).
35) Bayoumi, M. A. ed.: VLSI Design Methodologies for Digital Signal Processing Architectures, Kluwer Academic Publishers (1994).
36) De Micheli, G.: Synthesis and Optimization of Digital Circuits, McGraw-Hill Inc. (1994).
37) INTRODUCING TMS320C8x, Texas Instruments, Brochure (1994).

(平成6年8月23日受付)

宮崎 敏明 (正会員)

1981年電気通信大学応用電子工学科卒業．1983年同大学院修士課程修了．同年日本電信電話公社入社．以来，厚木電気通信研究所（現NTT LSI研究所）において，DSP用CAD, CADフレームワーク，論理設計支援技術および上位設計支援技術の研究開発に従事．1993年NTT光ネットワークシステム研究所伝送処理研究部に移り，プログラマブル伝送システム実現に向け，FPGAおよびFPGA用CADを中心とした研究に従事，現在に至る．主任研究員．工学博士（東工大）．IEEE，電子情報通信学会各会員．旧姓田中．