Journal of Zhejiang University SCIENCE A
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Optimizing pipeline for a RISC processor with multimedia extension ISA*

XIAO Zhi-bin (肖志斌)†, LIU Peng (刘 鹏)†‡, YAO Ying-biao (姚英彪), YAO Qing-dong (姚庆栋)

(*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: xzb@zju.edu.cn; liupeng@isee.zju.edu.cn

Received Mar. 20, 2005; revision accepted Aug. 12, 2005

Abstract: The 32-bit extensible embedded processor RISC3200 originating from an RTL prototype core is intended for low-cost consumer multimedia products. In order to incorporate the reduced instruction set and the multimedia extension instruction set in a unifying pipeline, a scalable super-pipeline technique is adopted. Several other optimization techniques are proposed to boost the frequency and reduce the average CPI of the unifying pipeline. Based on a data flow graph (DFG) with delay information, the critical path of the pipeline stage can be located and shortened. This paper presents a distributed data bypass unit and a centralized pipeline control scheme for achieving lower CPI. Synthesis and simulation showed that the optimization techniques enable RISC3200 to operate at 200 MHz with an average CPI of 1.16. The core was integrated into a media SOC chip taped out in SMIC 0.18-micron technology. Preliminary testing result showed that the processor works well as we expected.

Key words: Pipeline, RISC, Single-instruction-multiple-data (SIMD), Instruction set architecture (ISA), Multimedia extension
doi:10.1631/jzus.2006.A0269 **Document code:** A **CLC number:** TN402; TP37

INTRODUCTION

Embedded real-time multimedia applications that involve processing of video and audio streams demand an efficient media approach. A thorough survey of media approaches and architectures was given by Dasu and Panchanathan (2002). These media processing architectures can be classified into three categories including dedicated (application-specific) hardware, media processors and instruction set architecture extensions for general-purpose processors. As for embedded media systems, low cost and power with fast time to market is the top requirement. Based on the observation that many low price consumer products have relatively small chip area and well-defined workloads, configurable ISA extension to RISC processors is preferable for this kind of application (Ferreti, 2000; Dutt and Choi, 2003). However, many of such processors with multimedia extensions are not targeted at embedded market, as their main focus is PC or workstation characterized by high power consumption and high price (Lappalainen *et al.*, 2002). This is an important motivation of our research. In this work, we are interested in developing an extensible media architecture from a traditional RISC microprocessor to enhance DSP and multimedia stream handling capabilities. The approach makes possible a single-chip solution for some embedded multimedia applications (Ishiwata *et al.*, 2003; Liu *et al.*, 2005).

Compared with original Virgo core (Liu, 2001) in our laboratory, the new processor RISC3200 is designed to employ an SIMD based ISA extension approach for data-level parallelism. An extensible multimedia instruction set is developed to enhance multimedia processing capabilities of RISC3200. The introduction of multimedia instructions complicates the pipeline design which is the key technique for accelerating the execution of instructions for modern processors. The target operating frequency of

‡ Corresponding author

* Project supported by the Hi-Tech Research and Development Program (863) of China (No. 2002 AA1Z1140) and the Fork Ying Tong Education Foundation (No. 94031), China

RISC3200 is 200 MHz in a 0.18-micron technology. This paper discusses several pipeline related optimization techniques to achieve the speed specification and reduce the average CPI.

INSTRUCTION SET AND ARCHITECTURE

Multimedia instruction set

Multimedia and digital signal processing applications typically use short word data type (primarily 8- and 16-bit wide) and have more predictable behaviors and monotonously repetitive operations. The fundamental reduced instruction set (briefly MDF) of Virgo emphasizes few addressing modes and its simple processor architecture is not efficient for multimedia processing. By packing several small data elements into the 64-bit multimedia data-path, the multimedia extension instruction set (briefly MDS) of RISC3200 enables simultaneous processing of separate data elements. This form of SIMD parallelism is commonly known as sub-word parallelism (Lee, 1997). Our multimedia instruction set is similar to the existing media extension ISA in PC and workstation microprocessors. Besides general sub-word arithmetic instructions, permutation and saturation instructions are introduced to speed up operations like matrix transpose and special result handling in MDS. A PSAD instruction is proposed for minimizing the compute-intensive operations of video motion estimation algorithm by operating on eight pixels at a time. MDS also supports parallel multiplication and accumulation (MAC) operation commonly used in media processing.

System architecture

Fig.1 is a block diagram of RISC3200 architecture which works as a custom processor for a specific application. The extensible architecture combining the processing power of a multimedia unit with a 32-bit RISC core, capably handles both control and media processing. The architecture consists of a basic RISC core and an extension part. The RISC core includes a basic six-stage integer pipeline, separate instruction and data cache, on-chip data RAM, a DMA (Direct Memory Access) control module, a bus interface unit and a 32×32-bit general purpose register file. The extension part can have a multimedia extension unit, a memory management unit (MMU), an 8×64-bit wide media register file and other video/audio interface logic. The integration of memory management unit enables RISC3200 to support a full-featured embedded operating system. The extended pipeline including the basic integer pipeline and the pipelined multimedia extension unit is the candidate target for optimization in this paper.

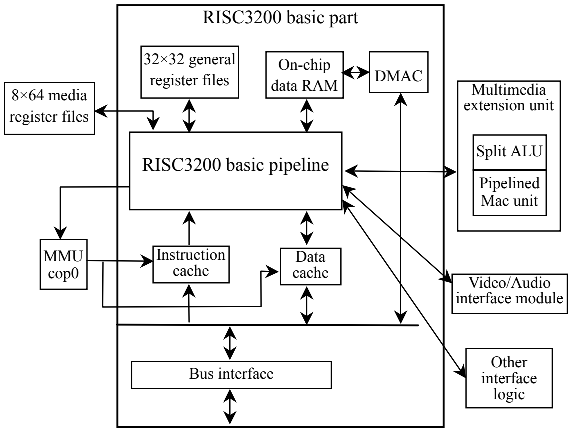

Fig.1 Block diagram of RISC3200 architecture

PIPELINE MODIFICATION AND OPTIMIZATION

While the RISC+SIMD architecture of RISC3200 makes it ideal for computational intensive tasks, a number of issues have to be overcome in the VLSI implementation of the processor especially on the design of the pipeline. This section will give a detailed discussion of the architecture and circuit optimization of RISC3200 pipeline.

Optimizing basic pipeline organization

The principle of pipeline design is to balance the length of each pipeline stage. Our goal is to limit the number of pipeline stages to avoid the reduction of IPC and fulfill the frequency spec. The Virgo pipeline is similar to DLX processor (Hennessy and Patterson, 2002), which includes a classic RISC 5-stage pipeline with two-phase clock scheme. The pipeline should be redesigned and optimized to support multimedia instructions and a single-phase clock scheme. Based on the standard-cell design method, the first optimization procedure is to analyze the critical path of the overall structure with the help of synthesis tools. By analyzing the synthesis results, critical path and signals on the path will be located. A data flow graph (DFG) with delay information is used to analyze the operation of the critical path. By breaking the critical path into different pipeline stages, the overall delay of each pipeline stage can be balanced and the frequency of the pipeline can be boosted.

The concept of DFG is shown in Fig.2a. Assume a module involves four operations including OP_{1a}, OP_{1b}, OP_2, OP_3 and the inputs of OP_3 come from the outputs of OP_{1a}, OP_{1b}, OP_2. Therefore, OP_3 has to wait for the completion of OP_{1a}, OP_{1b}, OP_2. We can assume that the operation delay is τ_{1a}, τ_{1b}, τ_2, τ_3 and that the wire delay is τ_{w1a}, τ_{w1b}, τ_{w2}, τ_{w3} correspondingly. Then the overall delay of the module is $\tau=(\tau_3+\tau_{w3})+\max(\tau_{1a}+\tau_{w1a},\ \tau_{1b}+\tau_{w1b},\ \tau_2+\tau_{w2})$. As OP_3 has to wait for the completion of other operations, then the overall delay of the module is dependent on the slowest operation of OP_{1a}, OP_{1b}, OP_2. In fact, we can break the operation OP_3 into several small operations such as OP_{3a}, OP_{3b}, OP_{3c}. We can assume that OP_{3a} is dependent on OP_{1a} and that OP_{3b} is dependent on OP_{1b} respectively. Then OP_{3a} and OP_{3b} can operate in parallel with OP_2 which is supposed to be the critical path of the module (Fig.2b). The overall delay of the module is reduced if OP_3 is carefully partitioned. The DFG with delay information can help us to locate the critical path of the original pipeline stage and guide our optimization work.

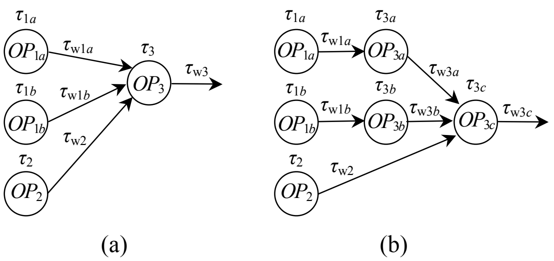

Fig.2 (a) DFG Mode I; (b) DFG Mode II

Because MDF and MDS instructions share the same split-ALU, the execution data-path is lengthened for additional store-alignment unit to support partial word store. The calculation of the virtual address and the store alignment operates in series, which makes the delay of the execution stage get close to 6.87 ns. Based on the DFG optimization method, the operation of the store alignment can be broken and proceeds in parallel with the virtual address calculation which can be put ahead to complete. Thus, the overall delay of the critical path in execution stage can be reduced to 5.02 ns.

RISC3200 supports virtual address. All the virtual addresses should be translated into physical addresses according to TLB (translation look-aside buffer) entries. The sequential access of TLB and cache makes the delay of IF and DM pipeline stage (Fig.3) longer. A virtual address indexed cache is used to enable access to TLB and cache can be performed simultaneously. Besides this, the data cache read and write operations are performed in DM and TC pipeline stage separately because the cache write signal is generated after the TLB access.

After a preliminary analysis and optimization, RISC3200 is designed to have a 6-stage pipeline organization as shown in Fig.3, including: IF (instruction fetching), ID (instruction decoding), EX (execution), DM (data memory access), TC (cache tag comparison), WB (write back).

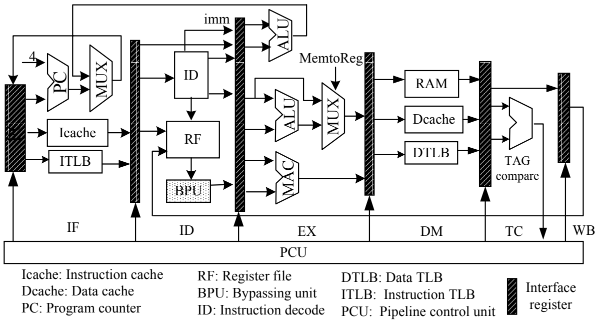

Fig.3 Pipeline organization of RISC3200

Scalable super-pipeline of EX pipe-stage

A scalable super-pipeline is utilized to extend EX pipeline stage to support the complex multimedia instructions. The "scalable" here means that different instructions require different clock cycles to complete in EX stage. The different execution delay of the MDS instructions will cause out-of-order execution. We propose a pipeline switch scheme to control the scalable pipeline in EX stage. A super-pipeline control module (Fig.4) is used to check the instructions flowing into EX stage. If the instructions may cause out-of-order execution, a SLIP signal would be sent to PCU to stop the instructions from proceeding; else the instructions select the appropriate entry to flow into the EX stage.

The DFG-based optimization principle is also

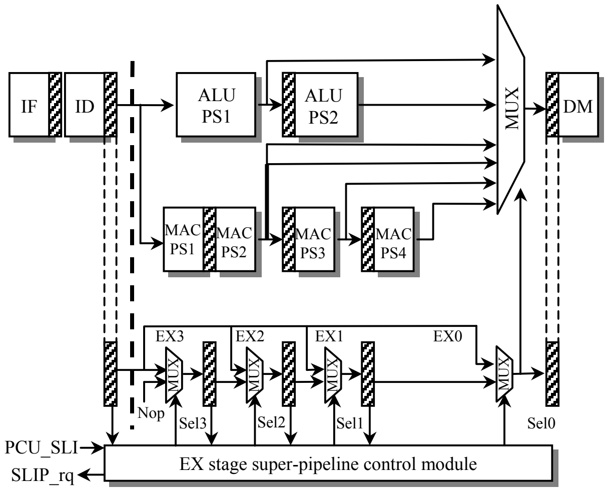

Fig.4 EX stage scalable super-pipeline

applied to optimize the scalable pipelined data-path in EX stage. We take the output results selection circuit optimization for example.

The EX stage results should be forwarded to the BPU or passed down to the next stage. An additional selection operation is required to just output a single result. The 8-to-1 multiplexer (MUX) lies in the critical path of the EX stage which requires 5.1 ns to complete without optimization (Fig.5a). Table 1 summarizes arrival time of 8 results from different function units in EX pipe-stage. We utilize the delay information shown in Table 1 to build a DFG analysis model. The analysis result instructs us to design a parallel MUX structure as Fig.5b shows. The overall delay of the readjusted data-path is reduced to 4.62 ns including the delay of the split-ALU and a single level of 2-to-1 MUX delay.

Table 1 Timing information of EX stage function units

ALU Results	Description	Arrival timing (ns)
Palu_dout	64-bit split ALU result	4.37
Psad_dout	8-way 8-bit PSAD instruction	3.21
Pmthiq	Register content transfer	0.40
Pmtloq	Register content transfer	0.40
Pmac16_ldout	Low 2-way 16-bit MAC	3.30
Pmac16_hdout	High 2-way 16-bit MAC	3.30
Pmadd_dout	Pmad instructions	2.51
Pmac32_dout	32-bit MAC instruction	3.94

Hazard handling

Pipeline hazards including structure, data and control hazards reduce the efficiency of the pipeline. Because of the simplicity of RISC3200 pipeline, only data and control hazards exist. The WAW (Write-After-Write) data hazard causes the out-of order execution of MDS instructions which is solved in the previous sub-section. We mainly focus on RAW (Read-After-Write) data hazards here. Because MDF and MDS instructions operate on different register files, RAW hazards arise from general purpose register and multimedia register hazard.

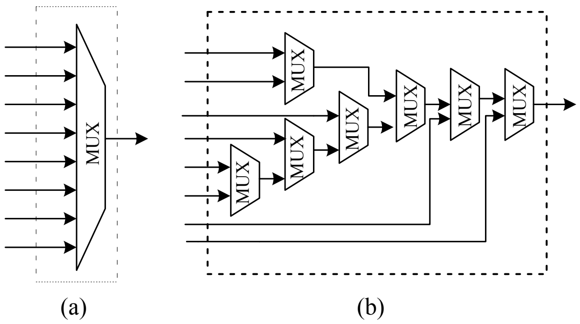

Fig.5 (a) 8-to-1 MUX; (b) Parallel MUX structure

A distributed bypass unit (DBPU) is designed to detect data hazards and forward data based on two principles: the destination register result should be forwarded to bypass unit as soon as possible; instructions should only wait for operands when they really need the operands. RISC3200 bypass mechanism includes forwarded data and control path. The bypass data path of RISC3200 is shown in Fig.6. The load result can be bypassed in TC and WB stage. All other ALU instruction results can be sent out in EX, DM, TC and WB stages. The results of multimedia instructions can be forwarded in the scalable EX pipe-stage.

Control hazards arise from the delay of fetching instructions and deciding next PC operation. Two-

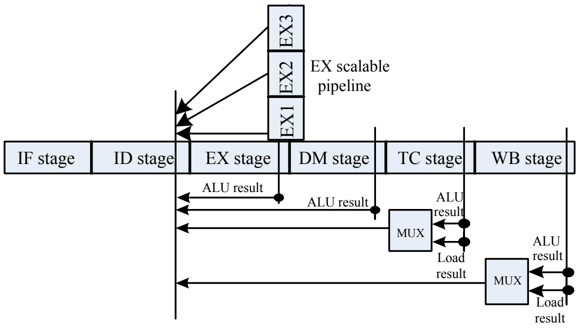

Fig.6 Distributed bypass scheme

cycle delay is needed as the calculation of next PC is arranged in EX stage (Fig.3). RISC3200 combines one delay slot and a static not-taken prediction scheme to solve control hazards. The scheme is simple to implement and appropriate for low cost embedded applications though the prediction accuracy is lower than that of dynamic prediction scheme.

Pipeline control unit optimization

Based on an FSM (Finite State Machine) scheme (Fig.7), a centralized pipeline control unit (PCU) is designed to act as the master controller of RISC3200 pipeline. PCU checks the pipeline state and signals from bus interface unit in every clock cycle. The pipeline state changes when events such as load data hazards, cache miss, exceptions, etc., happen and corresponding control signals are sent to every pipe stage immediately without being latched. Both the input and output signals are not latched in the PCU module. Therefore, the control signals will propagate through a long timing path, which limits further frequency increase of the pipeline.

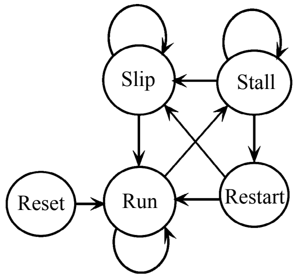

Fig.7 PCU operating mode

Fig.8 shows a typical timing path caused by the unlatched PCU control signals. The overall delay of the timing path is $\tau=\tau_1+\tau_2+\tau_3+\tau_4$, in which τ_1, τ_2, τ_3, τ_4 denote PCU request signal arrival time, PCU state transition time, PCU state decoding time and pipeline registers setup time respectively. In order to eliminate this long timing path, we partitioned the centralized PCU into two control units with manageable size. The partition is based on the priority and arrival time of PCU request signals.

The priority of each PCU request signal is based on the principle that signals from later pipe-stage should be responded to earlier. Table 2 summarizes the main PCU request signals and their arrival time. The RAW request signals lie in the middle of the longest combinational logic path, which starts from the ID module, through BPU and PCU modules to the pipeline registers. The delay of the path is about 5.45 ns which does not follow the clock spec. According to Table 2, GRF_RAW_rq, MRF_RAW_rq have the least priority and longest arrival time. Thus, another dedicated FSM is designed to accommodate the two RAW signals separately. Due to the intrinsic centralized control, additional MUX circuit is required to select control signals from different FSMs. With the partial centralized approach, the delay of the PCU timing path is reduced to 4.7 ns.

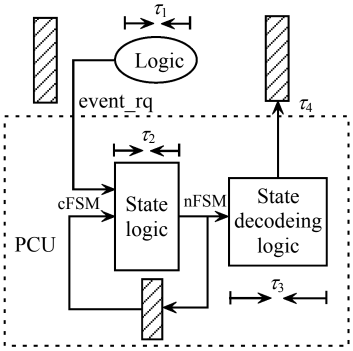

Fig.8 PCU timing path

Table 2 PCU request signals arrival time

Request signals	Stage	Signal description	Arrival time (ns)
ICM	ID	Instruction cache miss	1.30
GRF_RAW_rq	ID	General register RAW	2.09
MRF_RAW_rq	ID	Media register RAW	1.78
WAW_Slip	EX	Media register WAW	0.92
DCM	TC	Data cache miss	1.55
WTB	TC	Write buffer full	0.95
STP	TC	Store partial word	1.72

EXPERIMENTS AND RESULTS

RISC3200 has passed FPGA verification at XC2V3000 belonging to Xilinx Virtex-II series. The core is described in Verilog HDL except the on-chip memory which is based on SMIC synchronous SRAM library and implemented in 0.18-micron fabrication process. The core is synthesized by Synopsys Design Compiler. We use Synopsys PrimeTime to analyze the timing. Pipeline stage delay and average CPI are two main metrics for measuring the efficiency of the pipeline. After our optimization work, the synthesis result of RISC3200 is shown in Table 3. The delay of each pipeline stage only includes the combinational logic delay without the cache and TLB access time which exists in IF and DM pipe stage. The register files access time should be added to ID stage and delay of PCU should also be added to TC stage which may generate data cache miss signal. Considering the overall timing path of each stage, the delay is balanced and the critical path delay of the whole pipeline is 5.04 ns in the worst case.

Table 3 Synthesis result of RISC3200 pipeline

Module	Gate ($\times10^3$)	Delay (ns)	Module	Gate ($\times10^3$)	Delay (ns)
IF	4.5	0.53	GRF	30.9	1.70
ID	11.2	1.83	MRF	9.7	2.78
EX	90.9	5.02	JBU	3.4	2.41
DM	6.43	1.68	BPU	5.8	2.38
TC	4.8	2.40	PCU	0.9	1.79
WB	0.1	0.68	Pipe (all)	172.8	5.04

Note: SMIC 0.18 μm, 1.62 V, 125 °C

Based on an integrated software/hardware co-design development platform (Wu *et al*., 2004), both kernel DSP programs and real audio decoding programs have been carried out successfully by RISC3200 in the form of RTL dynamical simulation, FPGA prototype verification and real chip. As Table 4 shows, we have implemented a real-time MP3 (MPEG-1 Audio Layer-3) decoder on RISC3200 with average CPI of 1.15 (Yao *et al*., 2004). A radix-2 1024 points FFT (Fast Fourier Transform) program with CPI of 1.17 is also included in Table 4. The average CPI of the processor reaches 1.16 for the two test programs.

Table 4 Performance of RISC3200

Test programs	Code size (kB)	Ideal cycles	Real cycles	CPI
1024-point radix-2 FFT	1.62	262313	309524	1.17
MP3 Decoder (per frame)	18	1250678	1438235	1.15

CONCLUSION

This paper focuses on pipeline design and optimization for a 32-bit embedded RISC3200 processor with configurable multimedia extension instructions. The target operating frequency of RISC3200 is 200 MHz in a 0.18-micron technology. The RISC+SIMD architecture makes it ideal for computational intensive tasks and is especially preferable for low cost embedded consumer products. To achieve the speed specification and reduce the average CPI, several pipeline related optimization techniques including a DFG based analysis model are discussed in the paper. After our optimization work, we get an efficient pipeline with the critical path delay of 5.04 ns in the worst case and an average CPI of 1.16. RISC3200 was integrated into a media SOC chip which has been taped out in SMIC 0.18-micron technology. Preliminary testing result showed the processor works well as we expected.

References

Dasu, A., Panchanathan, S., 2002. A survey of media processing approaches. *IEEE Transactions on Circuits and Systems for Video Technology*, **12**(8):633-645. [doi:10.1109/TCSVT.2002.800866]

Dutt, N., Choi, K., 2003. Configurable processors for embedded computing. *IEEE Computer*, **36**(1):120-123.

Ferreti, M., 2000. Multimedia Extensions in Super-pipelined Micro-architecture. A New Case for SIMD Processing? Proceeding of 5th IEEE Int. Workshop Computer Architectures for Machine Perception, p.249-258.

Hennessy, J.L., Patterson, D.A., 2002. Computer Architecture: A Quantitative Approach, 3rd Edition. Elsevier Science Pte Ltd.

Ishiwata, S., Yamakage, T., Tsuboi, Y., Shimazawa, T., Kitazawa, T., Michinaka, S., Yahagi, K., Takeda, H., Oue, A., Kodama, T., Matsumoto, N., Kamei, T., Miyamori, T., Ootomo, G., Matsui, M., 2003. A single-chip MPEG-2 codec based on customizable media embedded processor. *IEEE Journal of Solid-State Circuits*, **38**(3):530-540. [doi:10.1109/JSSC.2002.808291]

Lappalainen, V., Hamalaine, T.D., Liuha, P., 2002. Overview of research efforts on media ISA extensions and their usage in video decoding. *IEEE Transactions on Circuits and Systems for Video Technology*, **12**(8):660-670. [doi:10.1109/TCSVT.2002.800865]

Lee, R.B., 1997. Multimedia Extensions for General-purpose Processors. Proceeding of IEEE Workshop Signal Processing Systems—Design and Implementation (SPIS'97), p.9-23.

Liu, P., 2001. Hardware/software codesign for embedded RISC core. *Proceedings of SPIE Media Processors*, **4674**:21-28. [doi:10.1117/12.451073]

Liu, P., Wang, W.D., Xiao, Z.B., Lai, L.Y., Teng, Z.W., Yu, G.J., Yao, Y.B., Chen, K.M., Jiang, Z.D., Zhang, Y.X., Zhou, J., Cai, W.G., Zhai, Z.B., Shi, C., Yao, Q.D., 2005. MediaSOC: A System-on-Chip Architecture for Multimedia Application. IEEE International Workshop on VLSI Design and Video Technology (IWVDVT2005), Suzhou, China, p.161-164.

Wu, H., Liu, P., Wang, W.D., Cai, Z., Yao, Q.D., 2004. Reconfigurable hardware/software cosimulation platform for media processor. *Proceedings of SPIE*, **5309**:114-122. [doi:10.1117/12.527231]

Yao, Y.B., Yao, Q.D., Liu, P., Xiao, Z.B., 2004. Embedded software optimization for MP3 decoder implemented on RISC core. *IEEE Transactions on Consumer Electronics*, **50**(4):1244-1249. [doi:10.1109/TCE.2004.1362526]