

EEMBC Office Automation and Networking Benchmarks on TX Series

Highlights

- EEMBC real-world benchmarks help designers select the right processor
- EEMBC members include more than 45 of the world's leading semiconductor, intellectual property, compiler and RTOS companies.
- Toshiba America Electronic Components, Inc. (TAEC) has run EEMBC benchmarks on many of its microcontrollers and microprocessors with outstanding results

Description

The Embedded Microprocessor Benchmark Consortium (EEMBC) develops and certifies real-world benchmark scores. These scores have become industry standards and are suitable for evaluating the capabilities of embedded processors and compilers according to objective, clearly defined, application-based criteria.

The EEMBC scores have replaced the one-size-fits-all Dhrystone benchmark by reflecting real-world applications and the demands that processors encounter in these environments. The result is a collection of "algorithms" organized into benchmark suites targeting telecommunications, networking, automotive/industrial, consumer, and office equipment products.

Features

- EEMBC has 6 benchmark categories with 42 benchmarks:
 1. Automotive/Industrial (16 benchmarks)
Engine control, in-vehicle communications
 2. Consumer (5 benchmarks)
Digital camera-like benchmarks
 3. Networking (3 benchmarks)
Workload of network routers
 4. Office Automation (4 benchmarks)
Printer-like benchmarks
 5. Telecommunications (6 benchmarks)
Modems, cellular, xDSL
 6. Microcontrollers (8 benchmarks)

- EEMBC scoring methods
 - Out-of-the-Box (ANSI C)
Compile portable code with any compiler and any compiler switches
- Optimized
 - Intrinsics/Language Extensions
 - Custom Libraries
 - Assembly Language
 - HW Accelerators

Supported Product Families

- All-embedded MPU and MCU
- Toshiba America Electronic Components, Inc., has run EEMBC benchmarks on many different MPUs and MCUs. Contact TAEC for the benchmark scores.

Specifications

Office Automation Specifications

- **Bezier Curve Calculations.**
Printer-page description languages and font-management routines typically represent curves as Bezier functions. This benchmark evaluates a series of Bezier curves, each with two endpoints and two control points. This benchmark exercises the mathematical computation capability of the target processor. Calculating points along a Bezier curve relies heavily on multiply and add operations. While the end points and control points for a Bezier curve are usually represented by floating-point numbers, it is not always necessary to use floating-point operations to evaluate the curve. A compile-time option allows this benchmark to use either floating- or fixed-point arithmetic during the parametric evaluation portion of the algorithm.

- **Dithering.**

The Dithering benchmark is representative of color and monochrome printer applications. The algorithm converts a grayscale image into a form ready for printing using the Floyd-Steinberg Error Diffusion dithering algorithm. This algorithm propagates an error quantity from image row to image row, effectively diffusing errors from the rendering calculations and preventing unwanted printing artifacts, such as banding. The benchmark uses two image buffers (one for the source image and a second for the generated output) and two line buffers to hold error data. The benchmark effectively stresses four areas of the target CPU: its indirect references, used for managing internal buffers; its manipulation of large data sets, since large images will stress the cache; its ability to manipulate packed-byte quantities, which are used to hold grayscale pixel information; and its ability to perform four-byte-wide multiply-accumulate operations per pixel.

- **Image Rotation.**

This benchmark uses a bitmap rotation algorithm to perform a clockwise, 90-degree rotation on a binary image. The bitmap rotation algorithm is primarily aimed at testing the bit manipulation, comparison, and indirect-reference capabilities of the microprocessor. The algorithm uses a series of indirect references and bit masks to check and set individual bits in a data buffer representing a binary image. The implementation supports 8-, 16-, and 32-bit data as well as little- and big-endian memory architectures. Two buffers are used, one for input and one for output, rather than trying to rotate the image in place.

- **Text Processing.**

The Text Processing benchmark is representative of a printer application. The algorithm parses Boolean expressions represented as text lines made up of variables, constants, and operators. The variables are from 1 to 64 characters long, the constants are single character "T" or "F," and operators may be either single-character symbols or their phonetic equivalents (and, or, etc.). Standard precedence rules for expression parsing apply. The expression is broken down into a binary tree structure, with each branch on the tree being an operand (a single variable, or a

constant, or a reference to yet another tree node representing another expression). Unary operators are stored as modifiers to each of the branches. The resulting structure is then traversed to evaluate the value of the expression. This benchmark exercises the byte manipulation, pointer comparison, indirect-reference handling, and stack-manipulation capabilities of a processor.

Networking Specifications

All the Networking benchmarks simulate the network activity under the CPU. All the packets do not go out to the real network, and no Ethernet connectivity is required to run these benchmarks.

- **OSPF (Open Shortest Path First)/Dijkstra.**

The OSPF/Dijkstra benchmark implements the Dijkstra shortest-path-first algorithm, which is widely used in routers and other networking equipment. The Dijkstra algorithm finds the shortest or least-cost path from a specific router (called the "source") to all other routers the source knows about. It builds a table of nodes where each node is a router. Each node has one or more arcs where each arc is a directed (one-way) link to another node. The arcs represent links between routers. Each arc has a cost value, which represents the "value" of the link. The lower the cost number, the more desirable it is to use that link. The Dijkstra algorithm starts at a source (or root) node. It then computes the best-case cost, or the shortest route (depending on how you look at it), of all the other nodes in the network in relation to the source node. There are two tables, arc_base and node_base. Each table is allocated one time before the benchmark starts (before any iterations). These tables are reinitialized after each iteration of the benchmark so that each iteration does exactly the same thing. Instead of building a predefined route, the standard method in this benchmark builds the routing tables dynamically.

- **Patricia.**

This EEMBC benchmark algorithm is a distillation of the fundamental operation of IP datagram routers: receiving and forwarding IP datagrams. All routers keep a table that allows them to look up an IP address and determine to which port an incoming IP datagram should be forwarded. Of course, the operation of a router is very complex, and this benchmark does not simulate a full router. Rather, it implements an IP-lookup mechanism based on a special data structure called a Patricia Tree (or "trie"). The Patricia tree data structure is a type of binary compact tree that allows fast and efficient searches with very long or unbounded length strings. This data structure is often employed in database searches. However, it is easily and efficiently applied to routing lookups. The number of search steps is bounded by the length of the search key (e.g., 32

Backgrounder

bits for IP version 4 addresses). The benchmark does build a tree from the IP address data in the file ROUTES.TXT. After the tree is initialized, the benchmark calls the function pat_search() for each IP address in LOOKUPS.TXT. One pass through this data is a single iteration.

- **Packet Flow.**

The Packet Flow benchmark implements some of the processing required in RFC1812, "Requirements for Routers." The benchmark uses two descriptor queues. One queue is called the receive queue (rx_queue in the code), and the other queue is the holding queue (hold_que in the code). The idea is to emulate how IP datagrams are often stored in actual systems using descriptors that are separate from the datagram. A descriptor has a next member, which allows it to be put in a linked list, and a pointer to a datagram. A

datagram is the IP portion (header and data) of a link-layer packet. This benchmark assumes that the link-layer packet has already been received and processed. For example, for an Ethernet interface, this would mean that the packet had been received with a valid Ethernet CRC with a uni-cast address. As each datagram is received and processed by the benchmark algorithm it is removed from the receive queue and placed in the holding queue. A single iteration of the benchmark is complete when the receive queue is empty. At the end of the iteration, the receive queue and the holding queue are switched, allowing the next iteration to execute with a full receive queue.

Related Tools and Systems

All the compilers used by the embedded MPU and MCU

Tables

Following is a table listing all the available EEMBC benchmark scores on Toshiba's MPU and MCU.

EEMBC Benchmarks

Toshiba MPU/MCU	Micro-Controller	Automotive	Consumer	Networking	Office Automation	Telecom
TMP86CM25F	▲					
TMP94FD53F		■				
TMP95FY64F					●	
TMP1940FDF					■	
TMPR3927F				●	●	
TMPR4926XB			■	■	■	■
TMPR4927ATB			●	●	●	●
TMPR4937			■	■	■	■
TX7901				■	■	

Legend: ■ benchmark score is available internally
 ▲ benchmark score is certified by EEMBC, but not published
 ● benchmark score is certified and published by EEMBC

benchmark score is available internally

Backgrounder

TAEC Regional Sales Offices

NORTHWEST

San Jose, CA

TEL: (408) 526-2400

FAX: (408) 526-8910

Portland, OR

TEL: (503) 629-0818

FAX: (503) 629-0827

SOUTHWEST

Irvine, CA

TEL: (949) 455-2000

FAX: (949) 707-5576

Richardson, TX

TEL: (972) 480-0470

FAX: (972) 235-4114

CENTRAL

Deerfield, IL

TEL: (847) 945-1500

FAX: (847) 945-2902

NORTHEAST

Wakefield, MA

TEL: (781) 224-0074

FAX: (781) 224-1096

Edison, NJ

TEL: (732) 248-8070

FAX: (732) 248-8030

SOUTHEAST

Duluth, GA

TEL: (770) 931-3363

FAX: (770) 931-7602

www.Toshiba.com/taec

The technical data may be controlled under U.S. Export Administration Regulations and may be subject to the approval of the U.S. Department of Commerce prior to export. Any export or re-export directly or indirectly, in contravention of the U.S. Export Administration regulations, is strictly prohibited.

TOSHIBA

TOSHIBA AMERICA ELECTRONICS COMPONENTS, INC.

**EEMBC Office Automation and Networking
Benchmarks on TX Series**

All trademarks are of their respective manufacturer and may be registered in certain jurisdictions.

© Copyright 3/2003 TAEC