



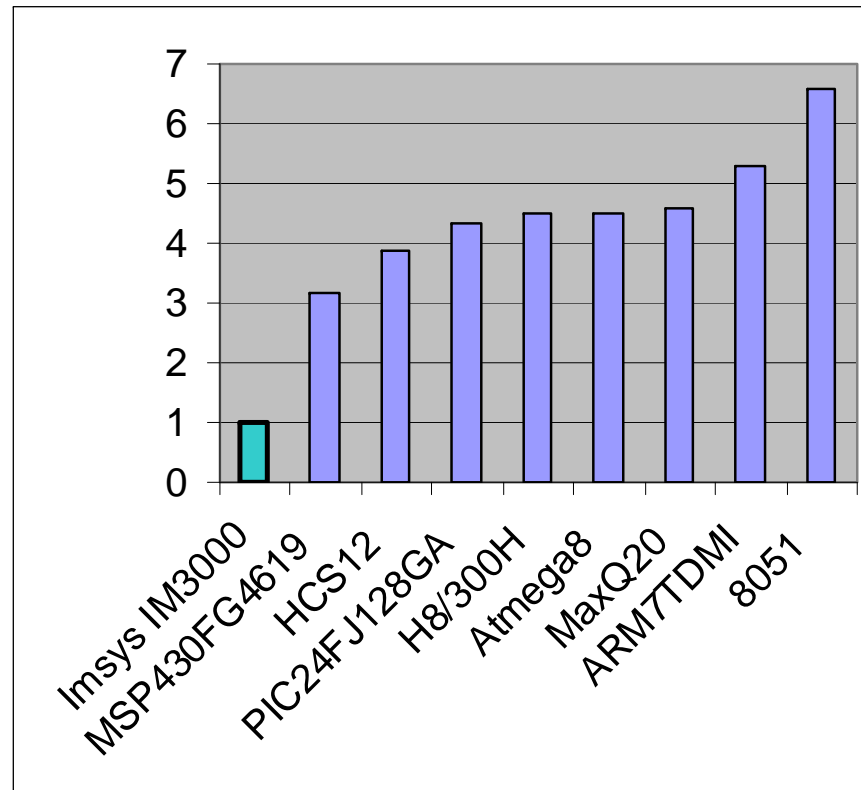
Imsys IM3000

Energy Efficient Processor
for Networked Embedded Systems



Code Size Performance

- Imsys IM3000
Clear Winner in
Non-Optimized
C-Code
- Compared
Against Both 8bit
and 32bit ARM
Processors

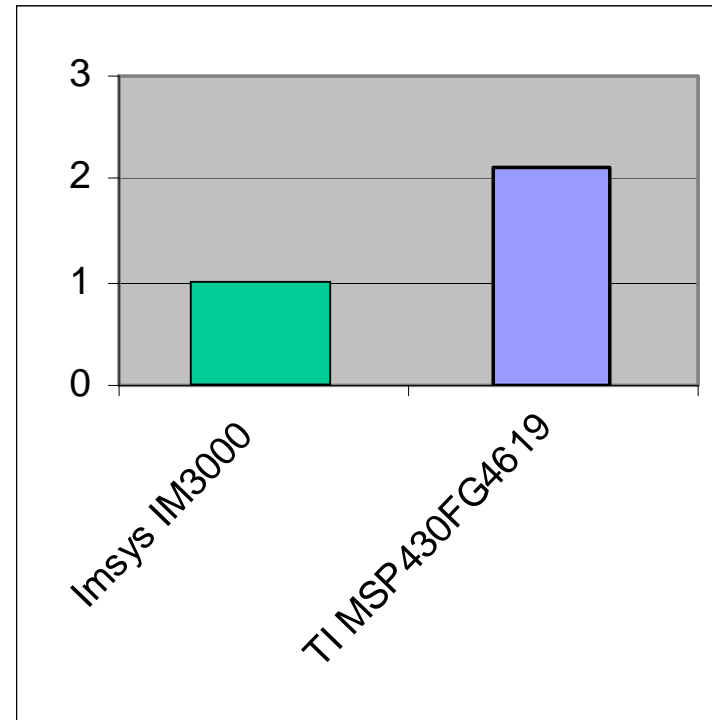


*ARM Uses Compact "Thumb" Instruction Set



Energy Consumption - with Unoptimized C Code

- Imsys IM3000 Twice as Efficient as the Presumed Efficiency Leader
- IM3000 Proved 5 Times Faster Using TI's Own Benchmarks

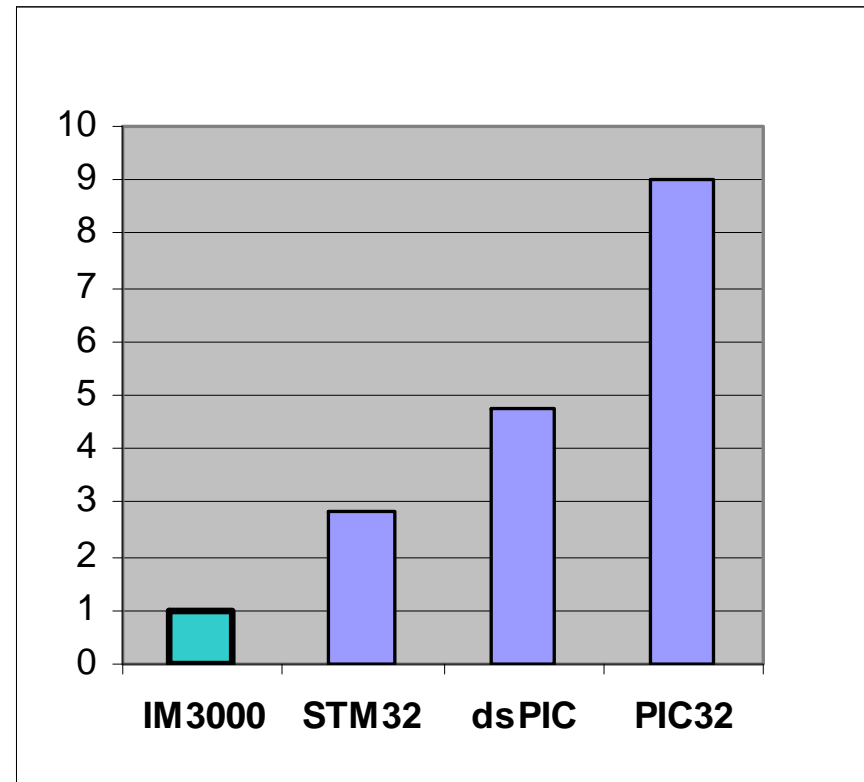


*C Code Benchmarks No Optimization



Energy Consumption - with Optimized Code

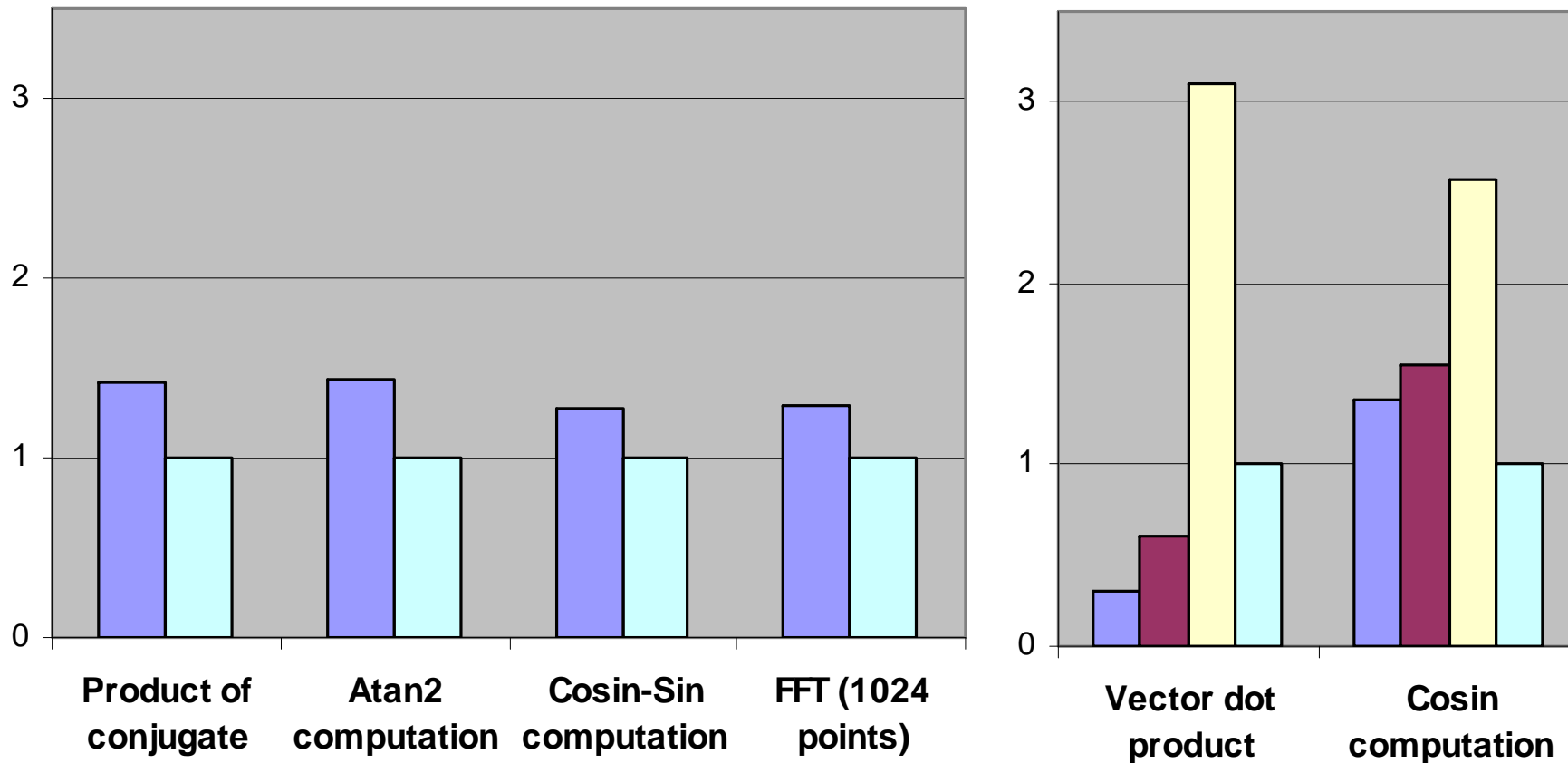
- Imsys IM3000 Clear Winner in Optimized Code
- Compared Against Two Most Popular 32-bit RISC Processors
- Compared Against Popular DSP Processor





Execution time

- Optimized DSP functions

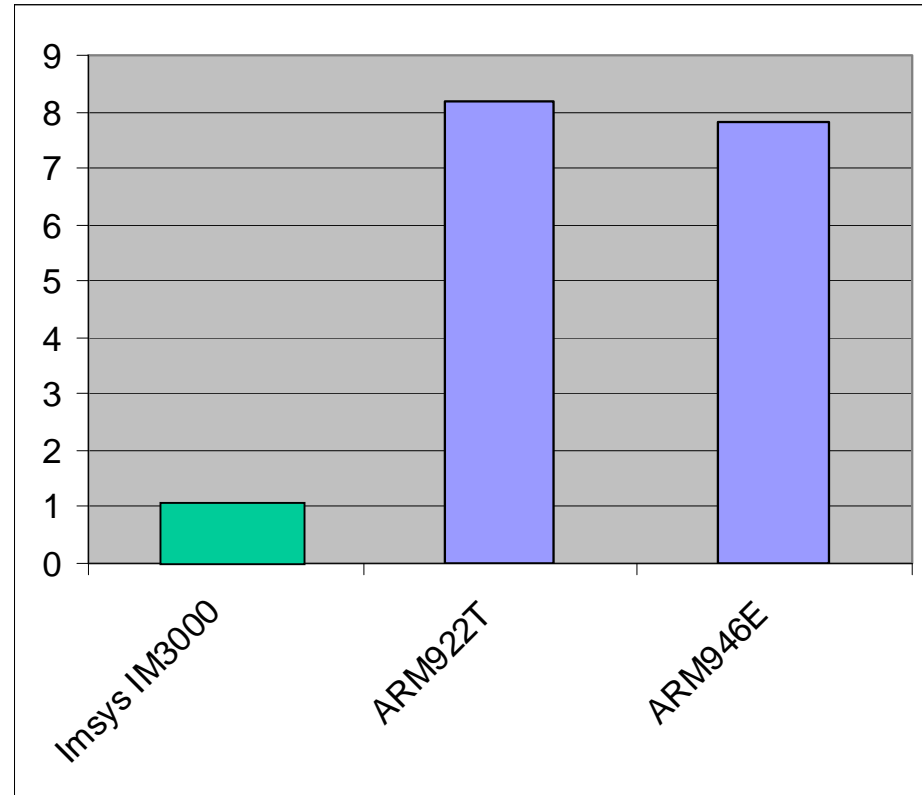


■ dsPIC ■ STM32 ■ PIC32 ■ Imsys IM3000



Power Consumption at Full Speed

- Imsys IM3000 Core vs. ARM9
- Execution Time Same Order of Magnitude
- Processors Utilize Same IC Fabrication Process

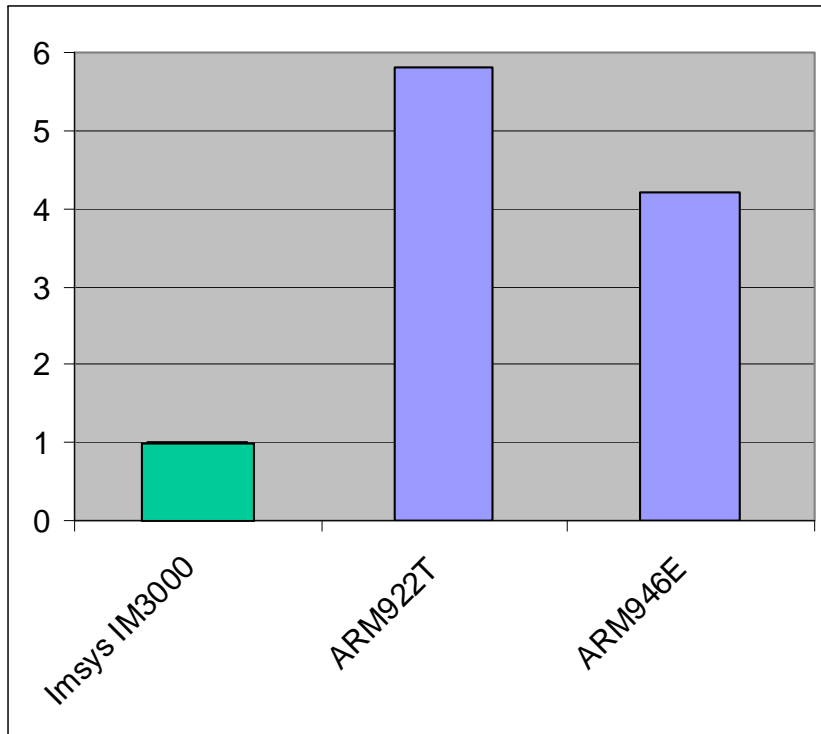




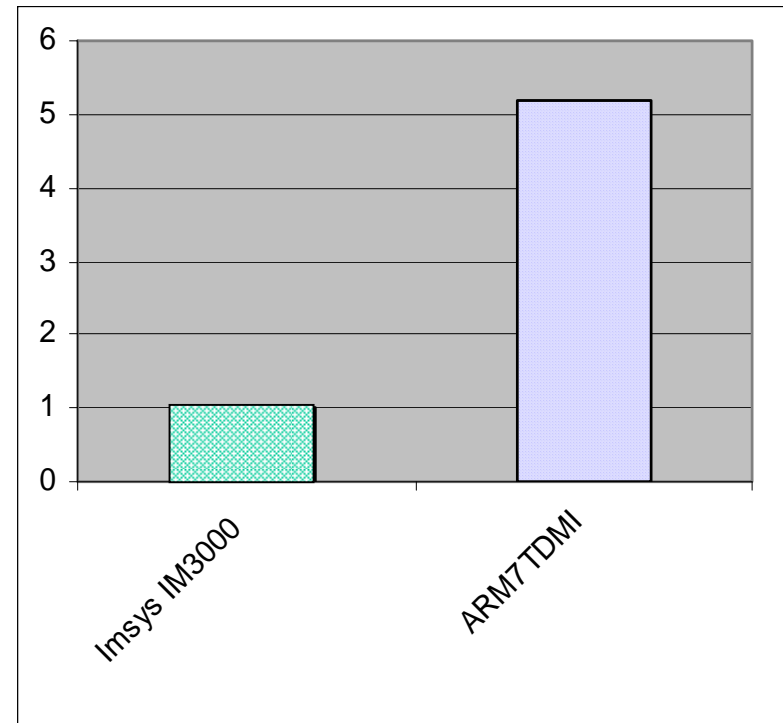
Size on Silicon

ARM vs. Imsys IM3000 Core

Processor Core Size



Program Memory Size





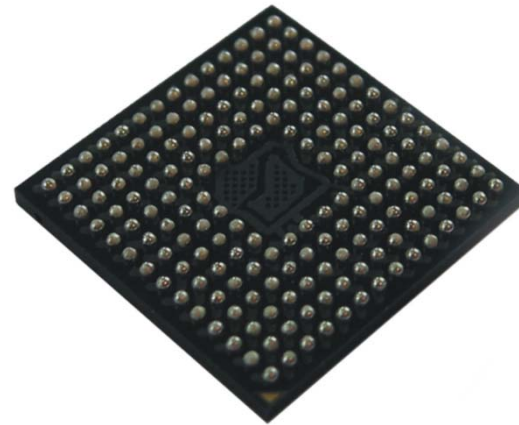
Sources

- Texas Instruments Benchmark Programs
- Non IM3000 Were Measured and Published by TI
- Published TI Cycle Count Results Used for MSP430
 - Some Information Obtained From Data Sheet
- Benchmark Programs and Results for dsPIC, PIC32 (MIPS), and STM32(ARM Cortex M3) Obtained from Independent Source
- Vendors Published Data Used for Power Consumption and Size of ARM Cores
 - Same IC Process Used for IM3000



IM3000 Optimization

- **Java**
 - JVM Bytecodes Execute Faster than ARM920T (Using Similar CMOS Technology)
- **Cryptography (ARC4, DES, AES, RSA, etc.)**
 - Efficiency Matching that of Dedicated Hardware
 - Much Faster than Other Small Processors
- **Digital Signal Processing**
 - Outperforms ARM and MIPS

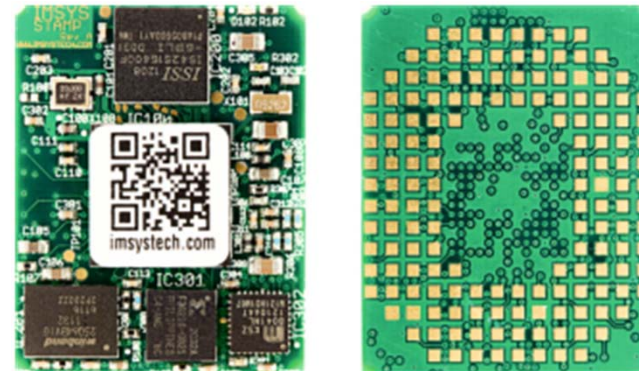




Modules

Velox

Compact LGA module
25x32 mm, 156 pads
IM3000, SDRAM, flash,
Optional Ethernet PHY



Aerius

Compact plug-in module
34x35 mm
IM3000, SDRAM, flash
3G+ cellular communication
Optional GPS



Imsys Developer

The screenshot displays the Imsys Developer IDE interface for a project named "im3000". The main window shows the source code for "main.c", which includes a main function that initializes DMA channels and starts the operating system. The code is as follows:

```
* The programs main function which does the following:  
*  
* 1) Initiates the DMA channels.  
* 2) Initiates the operating system, Rubus JOS.  
* 3) Creates the startup thread that will start up the system.  
* 4) Starts the operating system.  
*  
*****/  
int main(int argc, char *argv[])  
{  
    pthread_attr_t pthread_attr;  
    pthread_t th;  
    sysInitAttr_t sysattr;  
    josinit_attr_t jos_attr = {0};  
  
    // Init system. Setup DMA channels.  
    sysattr.dma[0] = 0;  
    sysattr.dma[1] = 0x400;  
    sysattr.dma[2] = 0;  
    sysattr.dma[3] = 0x300;  
}
```

The interface also features a Project View on the left showing the project structure, a Registers window on the right displaying hardware registers like PC, MSP, ESP, etc., and a Locals window showing variables like argc, jos_attr, pthread_attr, sysattr, and th. A debugger window at the bottom shows the execution timeline for threads such as SystemStartup Thread, josThreadIdle Thread, IP input Thread, and IPAddressTable Mutex. The debugger window also includes a Breakpoints table and a Memory window showing the current memory address and its contents.

File	Line	Function	Address
main.c	116	main	0x6616
HelloWorld.java	4	main(String[])	0x0

Address	Value	Comment
00000000	E4 63 FF FF FF FF FF 00 06	.c.....
0000000A	83 00 00 00 00 01 00 00 01
00000014	00 00 07 00 00 01 00 01 00 6AJ
0000001E	32 C9 00 00 00 00 00 00 00	2.....
00000028	00 00 00 00 00 07 F0 00 00 00
00000032	00 00 00 00 00 00 00 00 00
0000003C	00 00 00 00 00 00 00 00 00
00000046	00 00 00 00 00 00 00 00 00
00000050	00 00 64 B4 00 00 64 ED 00 00	..d...d..
0000005A	64 F0 00 20 00 00 00 64 00	d n d

Starting simulator
Reset done
Detected processor model: IM3000 (A)
Loading microprogram: im3210m11.mp (v0.1.1.0)
Loading application: im3000.gpx (address 00000000-0016258B)
Program arguments: "im3000.gpx -1"
Creating EPFS RAM disk...
RAM disk created, 1024KB
Controller successfully booted!