Application Note

AM263x Benchmark and Analysis for Traction Systems

Feng Qi

ABSTRACT

This application report contains benchmarks and analysis on typical application scenarios for the AM26x family of devices in traction inverters.

Table of Contents

List of Figures

List of Tables

Trademarks

1 Introduction

This application note presents AM2634 benchmarks obtained from a demo of the traction inverter framework, *AM263x for Traction Inverters*. The document also discusses typical application scenarios based on the benchmarks. The goal is to help users understand the performance of AM2634 in traction inverters.

System resources and diagram are presented in Figure 1-1 and Figure 1-2 for the following discussion. AM2634 is operated at 400 MHz core frequency and 200 MHz peripheral frequency, the program is compiled with TI Clang v1.3.0.LTS and O3 optimization for the best processing time, and the Field Oriented Control Interrupt Service Routine (FOC ISR) is assigned to FIQ of ARM Cortex R5F in order to achieve low latency. The performance of FOC ISR, FOC Run triggered by ADC INT1 in Figure 1-2, is the focus of the following discussion.

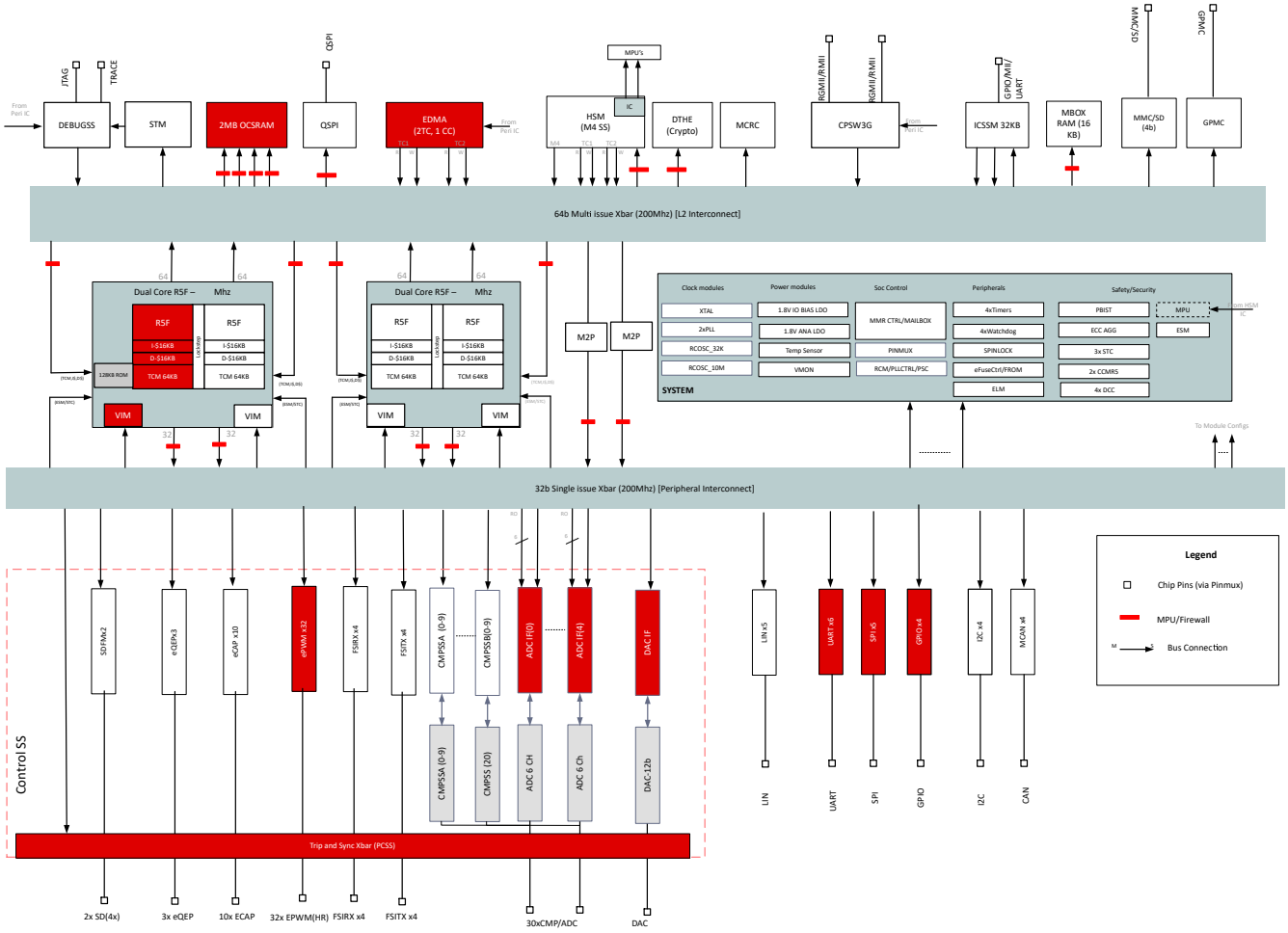

Figure 1-1. Resources in the Traction Inverter Framework

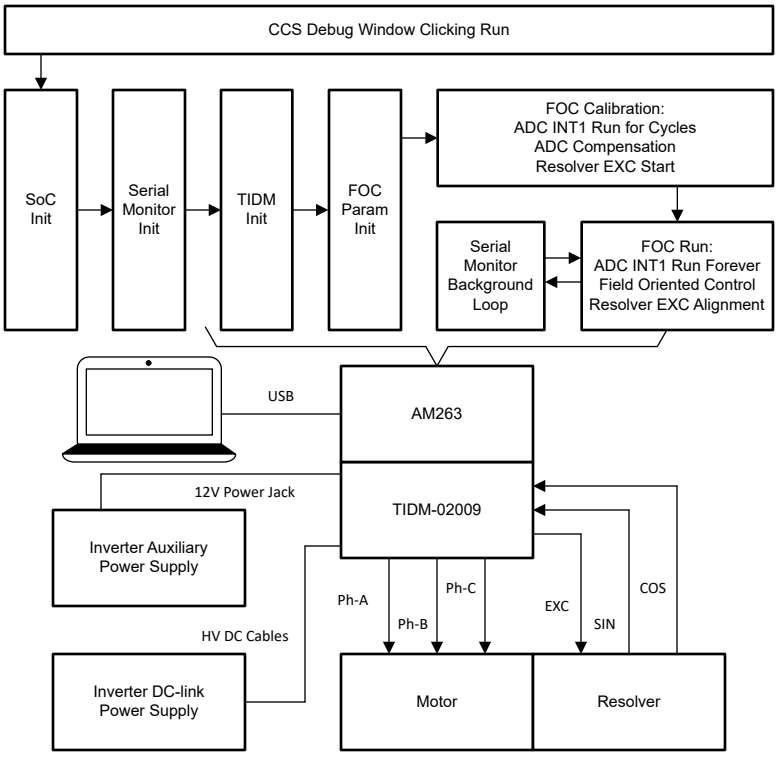

Figure 1-2. Traction Inverter System Diagram

2 AM26x FOC ISR Benchmarks

This section presents benchmarks of major functions in FOC ISR. They are visualized in Figure 2-1, Figure 2-2, Figure 2-3 and Figure 2-4. The order of items, top to bottom, in Bar Charts at Figure 2-3 and Figure 2-4 follows the execution flow. The Sunburst Charts in Figure 2-3 and Figure 2-4 are organized to have execution time from long to short in a clockwise direction. More details are summarized in Table 2-1. From Figure 2-4, the FOC ISR with TI R5F math library takes 3.9us from PWM time base counting Zero to PWM duty cycle updated. As shown in Figure 2-2, among the 3.9 µs, hardware interface costs 40% or 1.565 µs, and computation consumes 60% or 2.335 µs. Within the computation time, trigonometric functions take 26% or 600 ns. The Field Oriented Control excluding trigonometric functions and hardware operation is less than 1 µs, roughly 42% of computation time. The software resolver example without trigonometric functions requires 150 ns, 7% of computation time. The rest 25% computation time is spent on processing feedback and generating control reference in demo logic. The log function at the end of ISR is not counted in this benchmark as it is part of debug function.

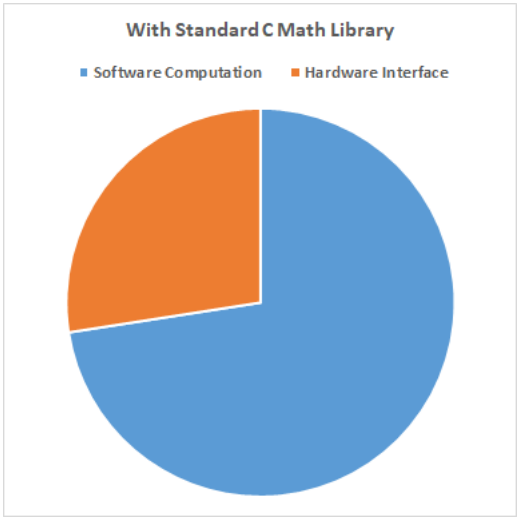

Figure 2-1. AM26x Software Computation vs Hardware Interface for FOC Loop With Standard C lib

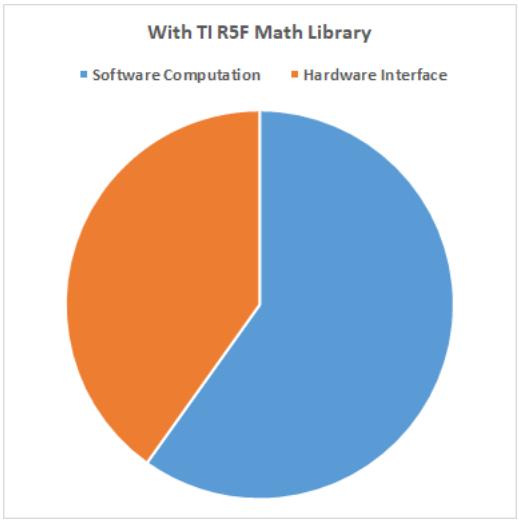

Figure 2-2. AM26x Software Computation vs Hardware Interface for FOC Loop With TI R5F lib

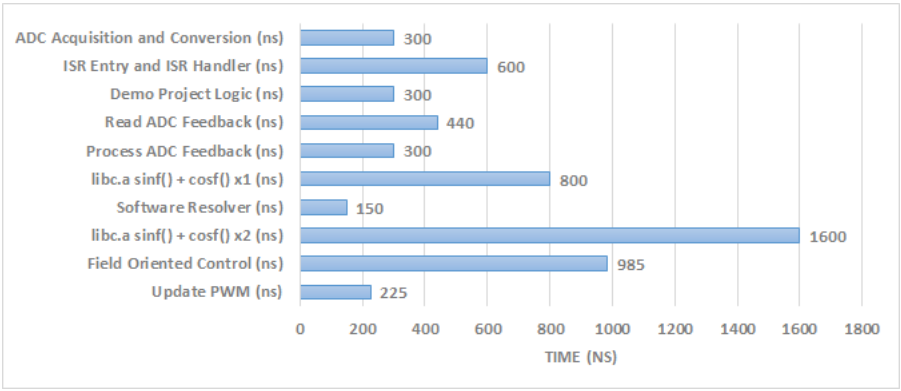

Figure 2-3. AM26x Time Consumption Breakdown for FOC Loop With Standard C lib

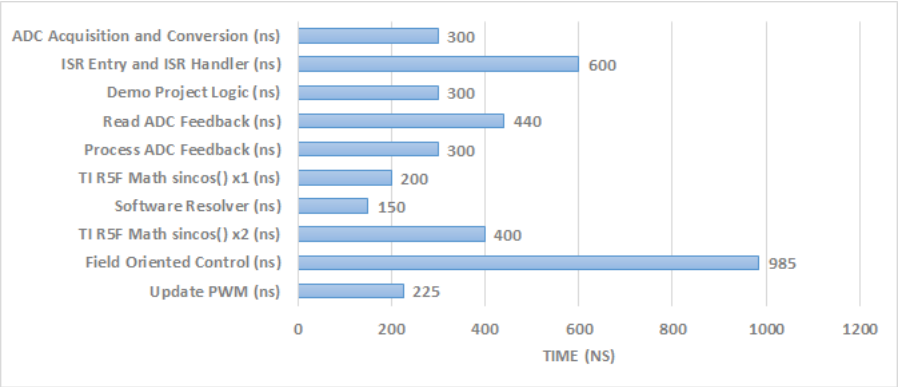

Figure 2-4. AM26x Time Consumption Breakdown for FOC Loop With TI R5F lib

Table 2-1. AM26x FOC Loop Benchmarks

Field Oriented Control Interrupt Service Routine	Description	Operation Type	Execution Time (ns)
ADC Acquisition and Conversion	1. PWM hardware trigger ADC 2. ADC hardware sample, convert, and trigger interrupt	Hardware Interface	300
ISR Entry	1. R5F hardware process interrupt 2. Interrupt handler load ISR	Hardware Interface	600
Read Feedback	1. Read 8 ADC results including 3-phase currents, resolver sin and cos, dc link voltage and others	Hardware Interface	440
Process Feedback	Remove effects, adjust scales, and filter noises	Computation	300
Run Trigonometric Functions - Option 1, Standard C Library	Run 3 times of sinf() and cosf() from libc.a 1. Run one time for software resolver 2. Run one time for Park transformation 3. Run one time for iPark transformation	Computation	2400

Table 2-1. AM26x FOC Loop Benchmarks (continued)

Field Oriented Control Interrupt Service Routine	Description	Operation Type	Execution Time (ns)
Run Trigonometric Functions - Option 2, TI R5F Math	Run 3 times of ti_r5fmath_sincos() from TI R5F Math 1. Run one time for software resolver 2. Run one time for Park transformation 3. Run one time for iPark transformation	Computation	600
Get Rotor Position with Software Resolver	resolver_run() is given as an example to take resolver feedback at the excitation frequency	Computation	150
Update PWM	Write PWM Duty Cycle	Hardware Interface	225
Field Oriented Control	1. Speed Loop Controller 2. Slip Compensation for Induction Motor 3. Park and Clark Transformation 4. Id Controller 5. Iq Controller 6. Output Limiter 7. iPark Transformation 8. Space Vector Generation 9. Duty Cycle Limiter	Computation	985
Demo Logic	1. Dummy theta generator for open loop and current loop tests 2. Test profile generator for speed loop test	Computation	300
Total	From PWM time base count Zero to PWM duty cycle update	Hardware Interface and Computation	3900

3 Typical Application Scenarios for Traction Inverters

There are many ways to design traction inverters but most share the similar ADC sample and PWM update scheme. This section provides two typical design examples for traction inverters operated in up-down counting or center-lined timing PWM. The cases are listed below. Unless new frequencies are specified in the following sections, the cases are analyzed in the condition of both PWM and resolver sinusoidal excitation frequency at 10 kHz.

- Sample ADC in cycle 0 and update PWM in cycle 1
- Sample ADC in first half of cycle 0 and update PWM in second half of cycle 0

3.1 Sample ADC in Cycle 0 and Update PWM in Cycle 1

As presented in Table 2-1, the FOC ISR takes 3.9 µs from PWM counting zero to PWM duty cycle updated. In a period of 100 µs at 10 kHz, the ISR only takes 3.9% of R5F at cluster 0 core 0, which is less than 1% time of all CPUs on AM263x. There are 99% of AM263x computation capability left to support other applications. But, it is highly recommended to keep control loop within one core as inter processor communication is time consuming, especially between different clusters.

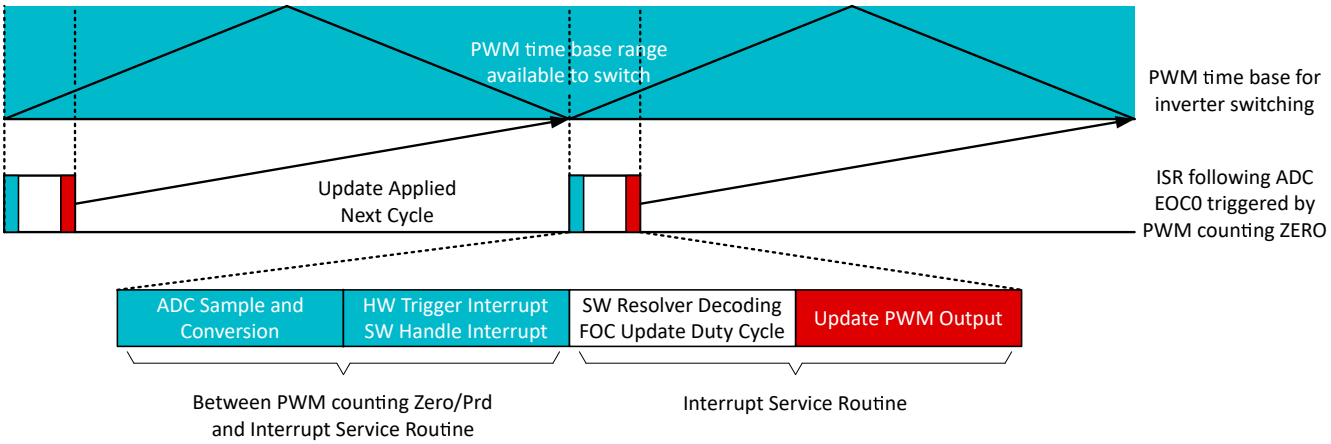

Figure 3-1. Sample ADC in Cycle 0 and Update PWM in Cycle 1

3.2 Sample ADC in First Half of Cycle 0 and Update PWM in Second Half of Cycle 0

In this case, feedbacks can be collected at 20 kHz with 10 kHz PWM. For the 10 kHz resolver, both peak and valley can be sampled and used after processing. The 3.9 µs excution time is about 7.8% of half PWM period, 50 µs. Similar to Figure 3-1, there is still more than 50% time from a single core left for customized operations if it is preferred to keep the time on the FOC ISR less than 60% of single core usage.

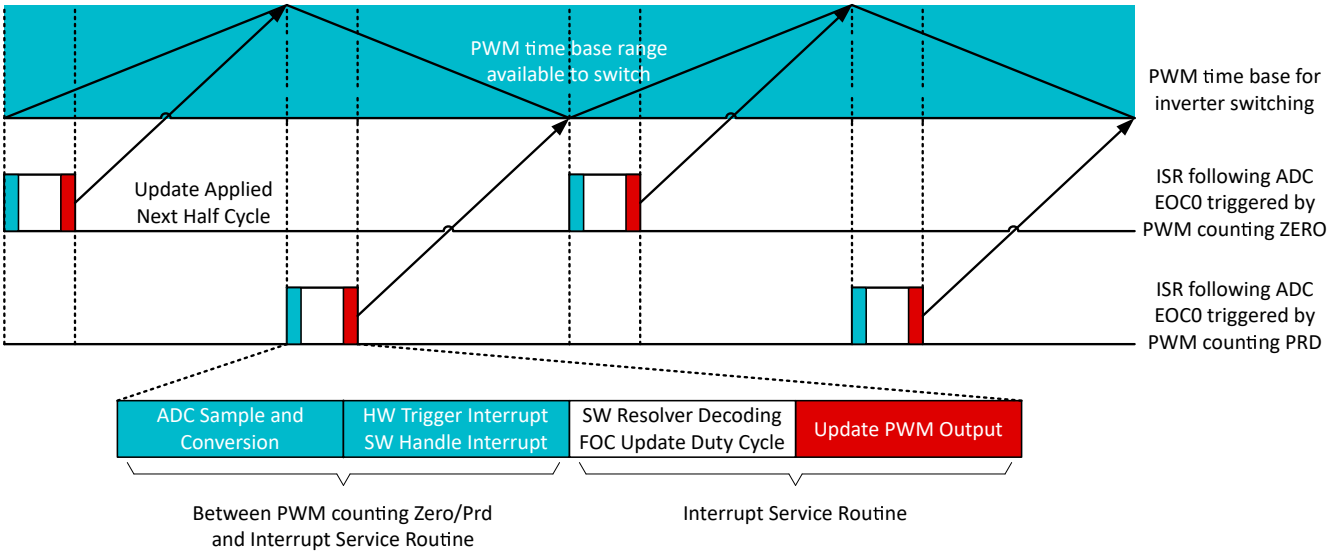

Figure 3-2. Sample ADC in First Half of Cycle 0 and Update PWM in Second Half of Cycle 0

On the other hand, in another case, 20 kHz PWM and 20 kHz resolver, the interrupt service routine will be operated at 40 kHz. 3.9 µs is about 16% of half PWM period, 25 µs. There is still more than 40% time if the ISR is limited by 60% of single core usage. If there is 40 kHz resolver, with 40 kHz PWM, the numbers can simply be scaled to 32% on basic control loop and 28% for customized operations. If it is 20 kHz resolver and 40 kHz PWM, the percentile on basic control loop should be reduced slightly by distributing resolver and speed loop among two cycles of current loop.

4 Summary

As discussed in previous sections, an AM263x traction inverter demo is optimized with O3 from TI Clang v1.3.0.LTS. It takes a 400 MHz R5F core 3.9 µs to run from the start of ADC sample to the end of PWM update. There are decent amount of time left on the same core for customized operations. And, there are another one R5F in the same cluster for dual mode configuration, and another cluster of R5F cores for either dual mode or lock step mode. If all cores are in dual mode, there are 400 µs CPU time available in a 100 µs PWM cycle. The 3.9 µs is event less than 1% of the 400 µs total CPU time.

Given the nature of traction inverter, the other cluster would probably be configured in lock step mode for AutoSAR, and other R5F in the same cluster could be used as a safety monitor. With this assumption, in the case of 20 kHz PWM, 20 kHz resolver, and updating twice in one PWM cycle, 3.9 µs is only 15.6% of the traction core usage and there is 44% of CPU time available if a 60% limit is set for control loop interrupt service routine in one core.

After all, AM263x family of devices is a powerful platform for traction inverters.

5 References

- Texas Instruments: *AM263x for Traction Inverters*

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265