Application Note AN-1128

IRMCK3xx One-Time Programmable Memory Considerations and Usage

By Benjamin Poiesz, International Rectifier
November 2006
Version 1.0

Table of Contents

This application note describes the OTP memory present in the IRMCK3xx series motor controllers. It also covers new features available to developers as well as some design considerations. Even though the IRMCK3xx controllers are pin-to-pin replacements for IRMCF3xx there are a few differences between those systems, which this application note covers. The methods of performing burn-in operations on the OTP are also reviewed, which follow the JTAG (IEEE Std 1149.1) specification.

1 Introduction

The OTP (One-Time Programmable) memory in the IRMCK3xx contains 64Kbytes of memory space that is split between the 8051 microprocessor and the MCE. 56Kbyes of that memory is reserved for 8051 program space, while the remaining allocation is used to load the 8K of RAM available in the system. This 8K is split into a 0.5Kbyte zone for the MCE Data (Dual Port Shared) RAM, 5.5Kbyte zone for MCE Program RAM, and 2Kbytes zone for 8051 local data RAM (Figure 1).

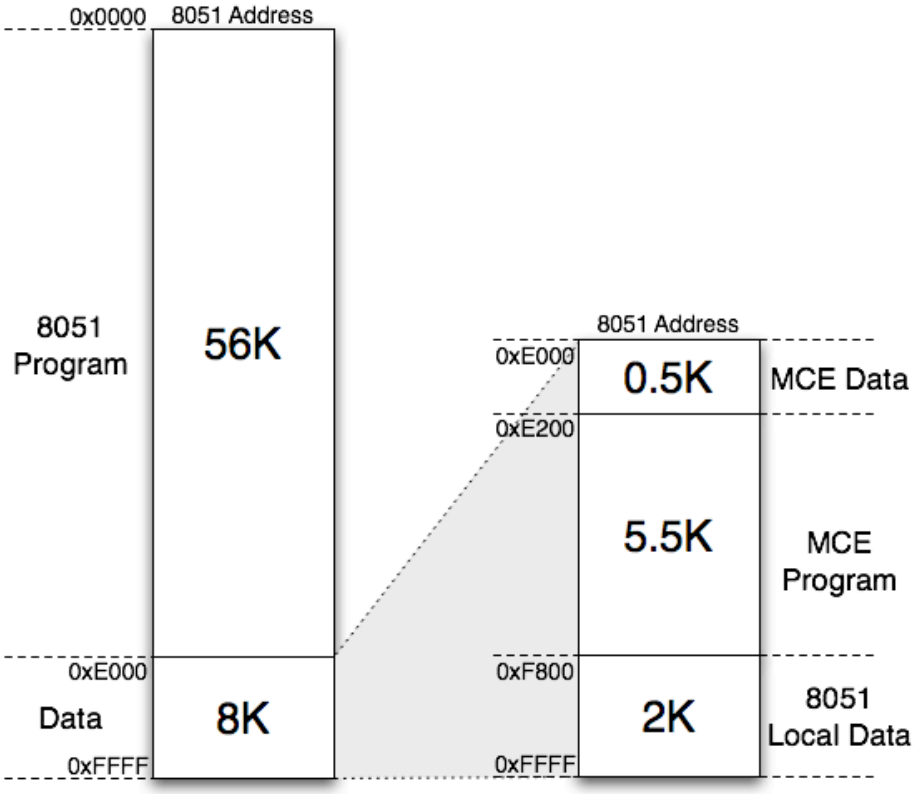

Figure 1. Memory Map of OTP

2 Bootload Sequence

When the controller is turned on, without a 8051 debugger connected, it will load the last 8Kbytes of OTP memory into the 8Kbyte RAM (see IRMCK3xx User Guide). This will effectively load the MCE program along with any shared initial data. If a debugger is connected the bootload process will not occur to prevent third parties from having access to the MCE program. The 8051 program space does not get transferred to RAM, as the OTP memory is directly read by the 8051 microprocessor.

3 OTP Memory Programming

3.1 Test Mode

The IRMCK3xx has a single JTAG port that can be used to either debug the embedded 8051 microprocessor or to program the OTP memory. To program the memory the controller must be put into 'Test Mode'. Once in this mode the OTP memory will be available over the JTAG interface.

3.2 Programming Pins

The OTP programming is performed over the standard JTAG interface available on the IRMCK3xx. In addition to this standard interface a supply voltage of 6.5V must be supplied to the OTP during programming. This supply voltage is supplied to a dual-purpose pin of SCL/VPP. When 6.5V is supplied to this pin it will act as the supply rail for the internal OTP memory. If performing programming in-circuit it is important to either verify that external components can withstand 6.5V or isolate the SCL/VPP pin during programming. An easy way to isolate these devices to have a jumper present on the SCL line; after the OTP is programmed the jumper can be put in place.

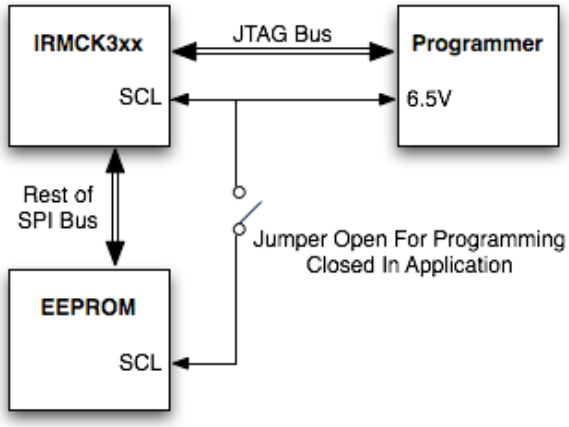

Figure 2. EEPROM Isolation Example

3.3 JTAG Overview

The JTAG interface in the IRMCK3xx is the standard four pin configuration. Data is shifted into TDI and shifted out of TDO. The state machine is controlled via the TMS line and TCK is the clock for communication. (Table 1)

Pin Name	TCK	TMS	TDI	TDO
Function	Clock	State Machine	Serial Data In	Serial Data Out
Direction	Input	Input	Input	Output

Table 1. JTAG Pins

3.3.1 TCK Cycle

Over the course of the JTAG programming cycle data should be loaded into TMS and TDI on the negative edge of TCK. TDO will change output states on the negative edge of TCK. Data will be sampled from the TMS and TDI lines on the positive edges of TCK.

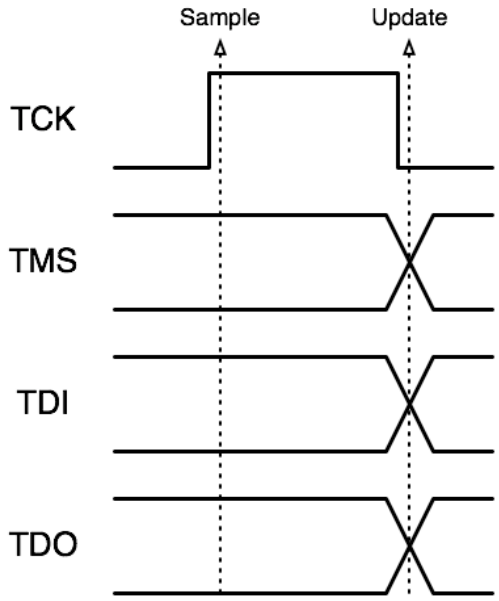

Figure 3. Basic JTAG Cycle

3.3.2 JTAG Registers

Two registers are present in JTAG implementations, IR (Instruction Register) and DR (Data Register). The JTAG interface will shift in and out the IR and DR registers, respectively, as it performs functions. To perform a command an instruction code is loaded into IR, and then the data associated with that command is loaded into DR. The order of IR and DR loads is dependent on the type of command being performed. There are additional registers defined for performing burn and verify operations listed below (Table 2). These registers can be written to and read from with specific IR values. It should be noted that the DR register is not a physical register, but can treated as one with respect to how it behaves in the system. More information can be found in IEEE Std 1149.1.

Register	Width	Description
IR	8 bits	Instruction Register
DR	16 bits	Data Register
otp_setup	8 bits	The OTP programming configuration register
otp_wr_timer	8 bits	Number of TCK clocks X 64 to write one byte of data
otp_jtag_address	16 bits	The current address of OTP that data will be burned to
otp_data	8 bits	The data byte that will be burned to OTP
test_modes	16 bits	Defines what mode the test interface is in (OTP Burn-in)

Table 2. The OTP Programming Registers

3.4 JTAG Register Descriptions

3.4.1 OTP_Setup[8:0]

The OTP_Setup register contains configuration information for accessing the OTP. It is defined as follows.

Address Advance			SKIP	POEB	PTM		
7	6	5	4	3	2	1	0

OTP_Setup.7 - OTP_Setup.5	AdAdv[2:0]	Number of address to advance in auto-increment modes (# to advance = 2^ OTP_Setup[7:5])
OTP_Setup.4	SKIP	Skip OTP_Addr[5:4]:0000 in auto-increment mode
OTP_Setup.3	POEB	OTP Output Enable Bit 1: Enables OTP memory read access 0: Must be zero for programming
OTP_Setup.2 – OTP_Setup.0	PTM[2:0]	Program Test Mode 000 = Read OTP 010 = 1x Normal Programming 011 = 4x Accelerated Programming

Table 3. OTP_Setup Register Definition

3.4.2 OTP_Wr_Timer[7:0]

This register adjust how long a OTP write operation should last. The write time is OTP_Wr_Timer[7:0] X 64 TCK cycles. This value then is dependent on the chosen TCK rate and minimum OTP write time.

3.4.3 OTP_JTAG_Address[15:0]

This register is the address that the next OTP data write command will go to. This is where the value written to DR will be written to OTP_JTAG_Address[15:0]. This register will auto-increment the OTP address after each write or read when OTP_Setup.4 is enabled.

3.4.4 OTP_Data[7:0]

This register is the data that the next OTP data write command will use to write to OTP_JTAG_Address[15:0]. This register is normally not accessed directly.

3.4.5 Test_Modes[15:0]

The Test_Modes[16:0] register defines what specific functions are to be performed by the test interface of the IRMCK3xx controller. Only one mode is needed for OTP programming, which is entered by writing 0x0002 into this register. This allows the TCK clock input to become the main system clock and, which allows synchronization between the control interface and the OTP memory.

3.5 IR Write Commands

IR Command	Command Description
0x00	Read JTAG TAP controller ID
0xF5	Enter the 'test mode' for OTP memory access
0xF6	Exit the 'test mode' and return to standard 8051 JTAG interface
0x70	Write contents of DR Test_Modes[15:0]
0x50	Write contents of DR to OTP_Setup[7:0]
0x51	Write contents of DR to OTP_Address[15:0]
0x52	Write contents of DR to OTP_Data[7:0]
0x54	Write contents of DR to OTP_Timer[7:0]
0x71	Enable auto-increment burn mode for OTP programming (see below)

Table 4. IR Write Command List

In order to perform these write commands the following actions should be taken.

Step 1:
Load IR
Step 2:
Load DR
Step 3:
System will auto-execute instruction with new DR

The 0x71 auto-increment command is slightly different in its operation. It exists to reduce the number of commands during programming. When 0x71 is set in IR every following DR load will cause that data to be written to the OTP and then automatically increment the OTP_Address[15:0] register. Therefore when programming the IR is set to 0x71 and then DR loads are repeated until the memory is fully programmed.

3.6 IR Read Commands

IR Command	Command Description
0x60	Read OTP_Setup[7:0] to DR
0x61	Read OTP_JTAG_Address[15:0] to DR
0x62	Read OTP_Data[7:0] to DR
0x63	Read OTP_Wr_Timer[7:0] to DR
0x64	Read OTP_Wr_Counter[15:0] to DR
0x72	Enable auto-increment read mode for OTP verification

Table 5. IR Read Command List

When the IR register is loaded with a read command it will return the specified register value to the DR register. This value can then be returned by reading DR and shifting it out over the TDO line.

3.7 TMS State Machine Diagram

This diagram describes is the standard JTAG state machine. Through use of only the TMS signal line either IR and DR values can be loaded into the system. For more information about JTAG refer to IEEE Std 1149.1.

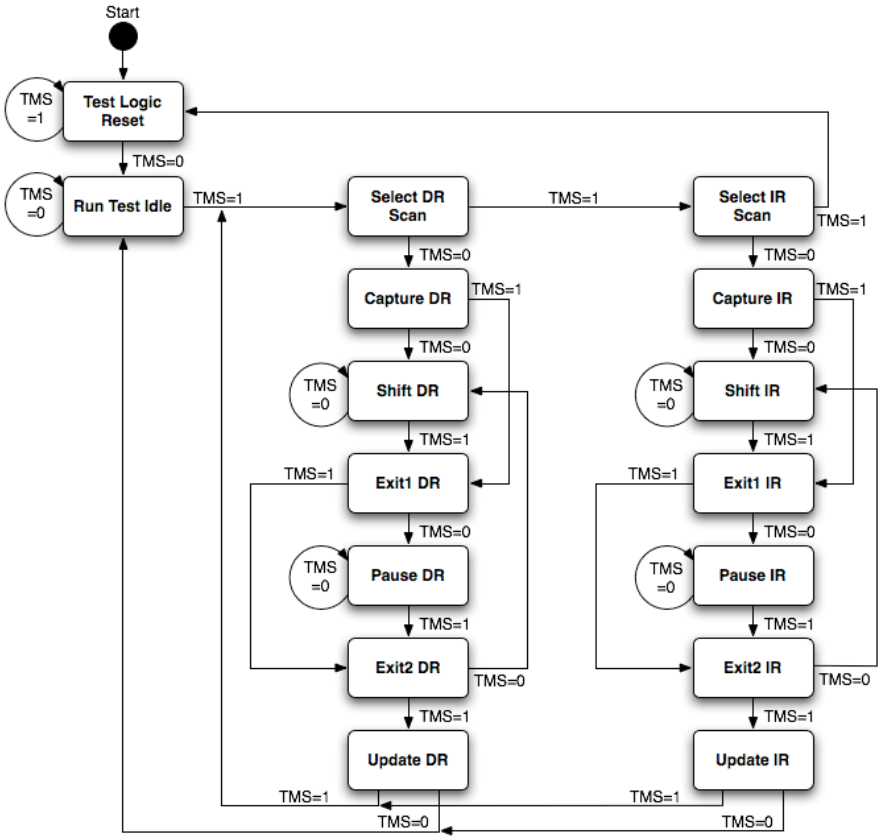

Figure 4. JTAG TMS State Diagram

3.8 OTP Programming Example

Write 0xF5 to IR	-	Enter test mode
Write 0x70 to IR then 0x0002 to DR	-	Set TCK to main system clock
Write 0x54 to IR then 0x07 to DR	-	Set OTP_Wr_Timer to 0x07
Write 0x50 to IR then 0x0A to DR	-	Set OTP_Setup to 0x0A
Write 0x51 to IR then 0x0205 to DR	-	Set OTP_Address to 0x0205
Write 0x71 to IR	-	Enable auto-increment OTP write
Write 0xA2 to DR	-	Write 0xA2 to OTP address 0x0205
Write 0xA3 to DR	-	Write 0xA3 to OTP address 0x0206
…		
Write 0xF6 to IR	-	Exit test mode

3.9 OTP Verification Example

Write 0xF5 to IR	-	Enter test mode
Write 0x70 to IR then 0x0002 to DR	-	Set TCK to main system clock
Write 0x50 to IR then 0x00 to DR	-	Set OTP_Setup to 0x00
Write 0x51 to IR then 0x0205 to DR	-	Set OTP_Address to 0x0205
Write 0x72 to IR	-	Enable auto-increment OTP read
Read DR	-	Dummy read
Read DR	-	Read OTP address 0x0205
Read DR	-	Read OTP address 0x0206
…		
Write 0xF6 to IR	-	Exit test mode

3.10 OTP IR Load Example

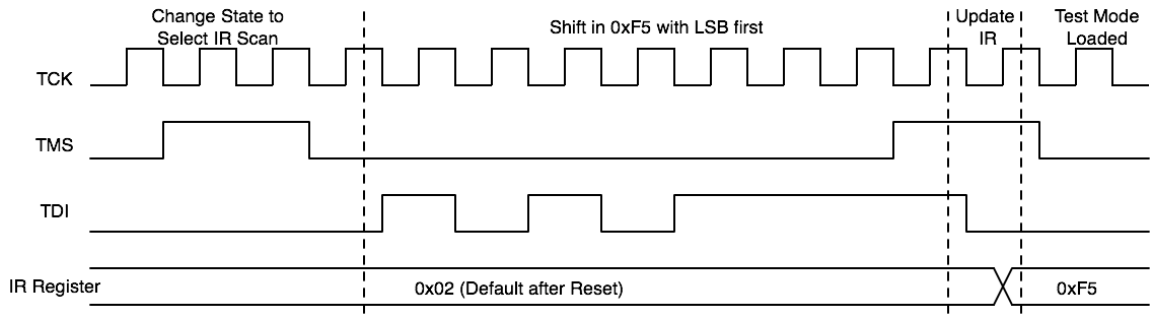

Figure 5. JTAG Load 0xF5 to IR (Enter Test Mode)

3.11 OTP Timing Information

There are three important parameters required when programming the OTP through the JTAG interface. This information is listed below.

Parameter	Symbol	Min	Max	Unit
Clock Cycle Time	T_{cyc}	25	-	ns
Program Pulse Width	T_{pw}	100	-	µs
Program Pulse Interval	T_{pwi}	5	-	µs

Table 6. Critical OTP Programming Timing Information

4 System Clock Frequency Limitations

Because the 8051 will execute its program directly from the OTP memory it must be constrained to the maximum operating frequency of the OTP. In the IRMCK3xx, the maximum OTP and therefore the 8051 frequency is 33Mhz. Because the MCE program is accessed via RAM it is possible to run the MCE significantly faster so long as the 8051's clock is divided.

5 System Clock Dividing

The IRMCK3xx supports running the MCE clock at multiples of 1, 2, 3, and 4 times the 8051 clock. This means the maximum frequency of 128Mhz could be maintained for the MCE, while the 8051 could run at 32Mhz. This would allow the OTP to operate below its maximum 33Mhz limitation, while having the maximum amount of computational power available in the MCE. This clock dividing is accomplished though two registers located at PLLF2[6:5]. These two registers will cause the clock divider to work in the following manner (Table 1). It is important to note that all of the UARTS, timers and all other clocked components except the MCE are on the slower clock.

PLLF2[6:5]	MCE Clock Frequency	8051 Clock Frequency
0b00	MstrClk	MstrClk/1
0b01	MstrClk	MstrClk/2
0b10	MstrClk	MstrClk/3
0b11	MstrClk	MstrClk/4

Table 7. PPLF2 Register Options For Clock Dividing

6 8051 Debugging

Even though the MCE program will not be transferred to RAM when the debugger is present it is still possible to debug designs. By having the 8051 manually transfer the MCE program from OTP to RAM at boot, debugging the full system is still possible.

6.1 OTP Memory Protection

To prevent third parties from connecting to the IRMCK3xx and downloaded proprietary motor control algorithms it is not possible to read the OTP contents from external sources such as a debugger. When reading the OTP from a debugger, the memory will be scrambled in a destructive manner. There is no way to recover the scrambled data that can be read from a debugger.

This scrambling is by default always enabled, and must be explicitly disabled during OTP programming. The very last byte of OTP (address 0xFFFF) determines if scrambling is enabled. If this byte is all 1s (0xFF) then scrambling is disabled. If this byte is anything else then scrambling is enabled. An additional step is required before JTAG can properly ready the OTP after setting this last byte to !0xFF. The JTAG interface must read this last byte. Upon reading this byte the controller will determine if the OTP will be allowed unscrambled read access.

6.2 Inserting Breakpoints

When debugging it is still possible to enter a breakpoint into the 8051 program, even though the memory is OTP. A single breakpoint can be entered by way of the FS2 debugger that will be trapped when set. This breakpoint must then be cleared so that a new breakpoint can be defined.

7 List of Figures

8 List of Tables