

LEON 3FT to S μ MMIT Interface

Table 1: Applicable Products

| Product Name | Manufacturer Part Number | SMD # | Device Type | Internal PIC* Number: |
|----------------------|----------------------------|------------|-------------|-------------------------------------|
| S μ MMIT E | UT69151_E | 5962-92118 | ALL | JA01 |
| S μ MMIT LXE/DXE | UT69151_LXE UT69151_DXE | 5962-94663 | ALL | MM016, MM025, MM027 |
| S μ MMIT XTE | UT69151_XTE | 5962-94758 | ALL | MM019, MM020, MM021, MM022 |
| LEON 3FT | UT699 | 5962-08228 | ALL | WG07 |

*Product Identifier Code

1.0 Overview:

The LEON 3FT Microprocessor (UT699) has emerged as a popular means for command and data handling systems for Hi-Rel applications. Providing the UT699 with the ability to interface a 1553 data bus can provide some design challenges. The UT69151 (S μ MMIT) is considered a standard in control and communications over a MIL-STD-1553 (1553) bus for many years. Currently the UT69151 is only available with a 5-volt interface, making it incompatible with the UT699 3-volt I/O. This Application Note provides a conceptual solution for interfacing the UT699 to the UT69151. The solution covers the methodology for implementing the interface in a Field Programmable Gate Array (FPGA), example state flow diagrams and Algorithm State Machine (ASM) diagrams for defining the required behavior between the UT699 and the UT69151.

2.0 Technical Background:

At times, a design must interface components with different I/O voltage requirements. One of the most versatile solutions for mating I/O of different voltage domains is a FPGA. Many FPGAs provide the versatility of selecting I/O voltages and drive strengths to provide voltage translation across many domains.

2.1 LEON 3FT (UT699):

The LEON 3FT provides control of systems through Compact Peripheral Component Interconnect (CompactPCI^{®1}), SpaceWire (SpW), Universal Asynchronous Receive and Transmission (UART), General Purpose I/O (GPIO), CAN 2.0 (CAN), and 10/100Mbps Ethernet. A host must have access to a MIL-STD-1553 device to receive and transmit messages across a 1553 bus. The LEON 3FT has the ability to access external devices from within the memory mapped I/O space. The control of the devices is generally augmented using the GPIO. The block diagram in Figure 1 shows the available interfaces for development of a system level project.

¹ CompactPCI and the CompactPCI logo are registered trademarks of the PCI Industrial Computers Manufacturers Group.

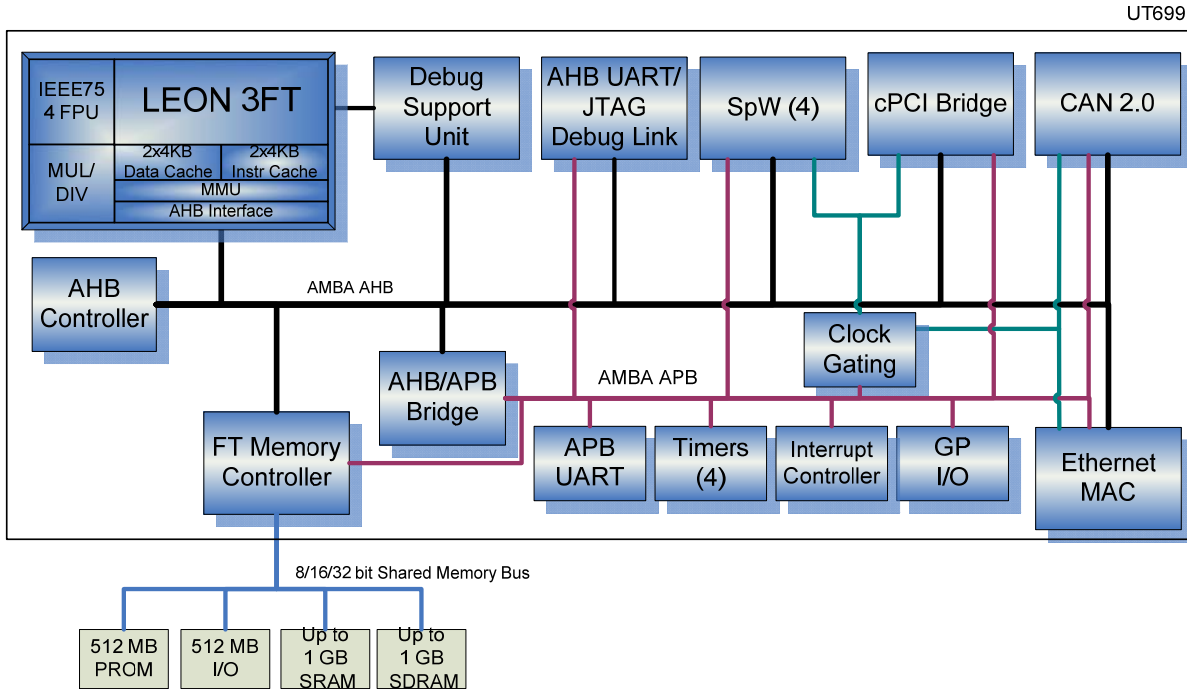


Figure 1: UT699 Block Diagram

2.2 UT69151 (S μ MMIT):

The S μ MMIT has the ability to transmit and receive 1553 protocol messages autonomously and provide the information to a host processor. The S μ MMIT has a long flight heritage and a multitude of features making it the preferred interface solution to a 1553 network. The S μ MMIT key features include operations as a Remote Terminal (RT), Bus Controller (BC), Monitor Terminal, RT/Monitor (RT), and Ping Pong mode. Figure 2 shows the interfaces available on the S μ MMIT.

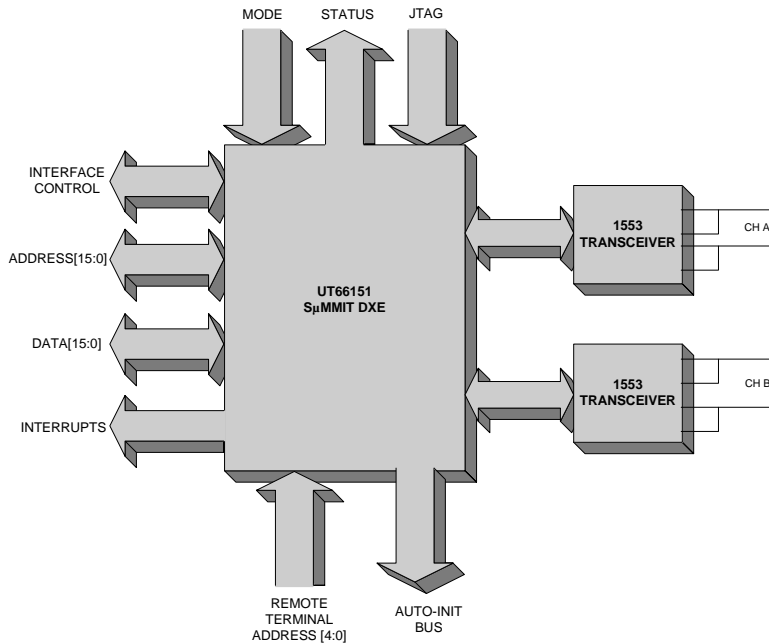


Figure 2: UT69151 DXE Block Diagram

3.0 Implementation of the FPGA Based Interface:

The LEON 3FT to SμMMIT System Diagram (Figure 3) shows the signals required from the LEON 3FT to the FPGA and signals from the SμMMIT to FPGA. The figure shows the division of the two I/O domains for the example FPGA. Figure 3 depicts the basic connections the FPGA requires for interfacing to each device (**NOTE:** connections may vary according to implementation).

3.1 Conceptual System Design:

The first block in the diagram, represents the LEON 3FT and I/O for interfacing to the FPGA. The main interfaces used on the LEON 3FT include the GPIO and Fault Tolerant (FT) Memory Controller. The GPIO lines provide secondary control and status feedback to and from the FPGA. The LEON 3FT uses these lines to determine status of the FPGA and SμMMIT.

The second block in the diagram represents a FPGA chosen to meet the project requirements. There are many FPGA’s available that vary in number of configurable blocks, storage elements, fault tolerance, and I/Os. Many projects require a FPGA that is fault tolerant such as the UT6325 (Aeroflex RadTol Eclipse FPGA) or the SX72 (Actel FPGA) with I/O voltage translation. The choice of FPGA is a consideration of the project requirements. The FPGA provides the I/O voltage translation between the 3-volt and 5-volt domains as seen in Figure 3. The FPGA also provides clock domain synchronization, logic control to each device interface, controls reads and write to the SμMMIT and shared memory; and prevents bus contention between the LEON 3FT and SμMMIT accesses to the shared memory.

The third block in the diagram represents the SμMMIT and I/O for interfacing the FPGA and shared memory (e.g. Aeroflex UT9Q512E 5-volt 4Mb SRAM (512Kx8)). The FPGA uses the interface controls to read and write to the SμMMIT internal registers and monitor operations related to 1553 bus command and control.

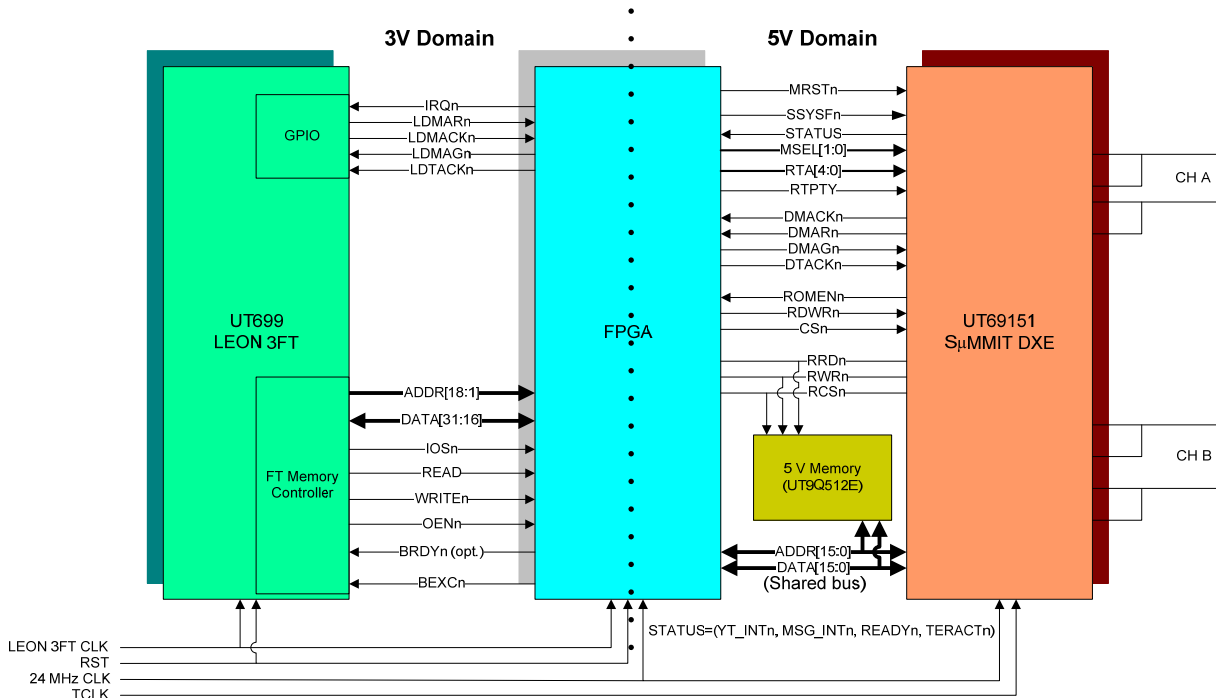


Figure 3: LEON 3FT to SμMMIT System Diagram

3.2 Conceptual Interface Design:

The example Interface Design block diagram shown in Figure 4 consists of an Interface Controller, Status Controller, DMA Controller, Data Path Controller, and a Configuration Register block. Each block provides a starting point for defining the interface and associate behavior.

The Interface Controller provides primary control of interactions between the LEON 3FT and the SμMMIT. The Interface Controller is responsible for control signals to the Data Path Controller to allow the LEON 3FT access to the SμMMIT registers and shared memory. The IC handles clock domain synchronization, reads and writes to the configuration register, control of the Status Controller, and contains address and data registers for processing LEON 3FT commands and data across a slower clock domain. The Configuration Register block is set of static registers to the SμMMIT and typically located inside of the Interface Controller, but shown here to emphasize the need for a registers for initial startup of the SμMMIT. This Interface Controller provides central control of operation for the other control units in the interface.

The SC provides real time monitoring of the SμMMIT interrupt and status signals, interface functional status, and providing a signal notifying the LEON 3FT of changes in critical status signals. The SC registers capture interrupt and SμMMIT status signals.

The DMAC provides bus arbitration and access for the LEON 3FT and SμMMIT. The DMAC provides the control signals to the Interface Controller and Data Path Controller permitting control over the shared Address and Data bus.

The DPC provides control and bias of all I/O to or from the LEON3 FT and SμMMIT. This DMAC controls the direction of the data passing through it, I/O buffering, I/O termination, and I/O pull-up/down requirements.

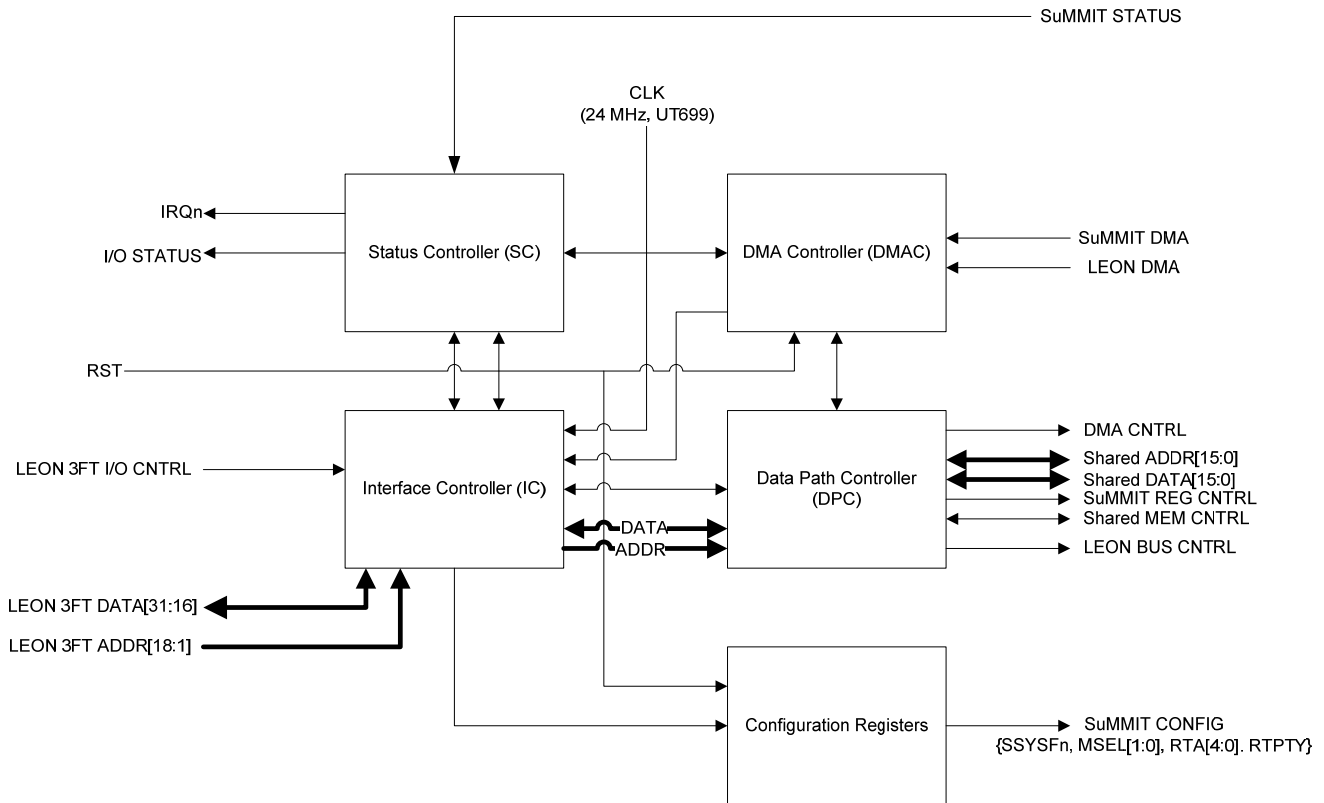


Figure 4: Interface FPGA Block Diagram

3.3 Behavioral Definitions:

Behavior definitions are typically done using a combination of flowcharts and Algorithmic State Machine (ASM) diagrams. A series of behavioral definitions defines the synthesizable logic. Each control block as shown in Figure 4, defines the ASM diagram or flowchart associated to it. The diagrams in Figure 5, Figure 6, and Figure 8 are examples of ASM diagrams; each diagram has states identifying a required behavior and each behavior should have an associated flowchart. Figure 7 and Figure 9 are examples of behavioral flowcharts.

3.3.0 Interface Controller (IC):

The IC is the main controlling entity in the design. The primary function of the IC is to synchronize signals across clock domains and determine which device to provide control and how each interaction will occur. This implementation of the controller handles reading and writing to the S μ MMIT, shared memory, and S μ MMIT startup configuration register. This controller provides read/clear control to the Status Controller based on LEON 3FT I/O control signals (IOS, OENn, READ, WRITEn, GPIO (LDMACKn, LDMARn)), and the bus operations (bus_op) signal. A description of each signal is listed in Table 2.

Table 2: IC Signal Description Table

| Signal | Signal Direction | Purpose/Description |
|------------------|------------------|--|
| RST | Input | Global interface reset |
| IOSn | Input | UT699 memory mapped I/O select to select FPGA |
| OENn | Input | UT699 memory mapped I/O output enable signal to enable memory output from the FPGA |
| READ | Input | UT699 memory mapped I/O READ signal to read from FPGA |
| WRITEn | Input | UT699 memory mapped I/O WRITE signal to write from FPGA |
| ADDR[17:16, 2:1] | Input | UT699 Address signals that decode to access the S μ MMIT registers, Shared Memory, Configuration registers, and Clear signal to the SC |
| LDMACKn | Input | UT699 GPIO signal to the DMA interface for acknowledgement of control of the shared bus |
| LDMARn | Input | UT699 GPIO signal to the DMA interface for request of the control of the shared bus |
| bus_op | Input | Interface Bus Operations signal indicating the use of the bus |
| bsy | Output | Interface Busy control signal to control duration of state transitions |
| stat_out | Output | IC Status Output signal to indicate the status of the task being performed |
| csb | Output | IC Chip Select control signal to the S μ MMIT and Shared Memory |
| rwb | Output | IC Read/Write Select control signal to the S μ MMIT and Shared Memory |
| doe | Output | IC Data Output Enable control signal to the S μ MMIT and Shared Memory |
| clr | Output | IC Clear control signal to clear the output of the SC |
| config | Input/Output | IC Read/Write of configuration registers for the S μ MMIT |

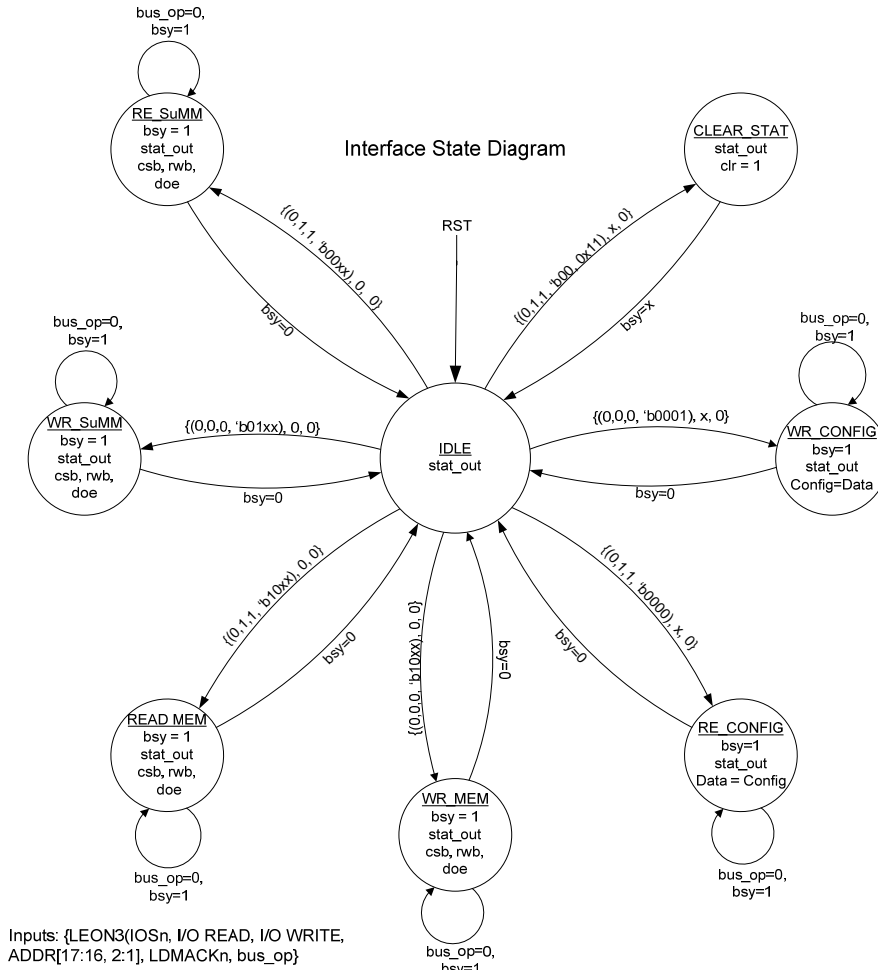


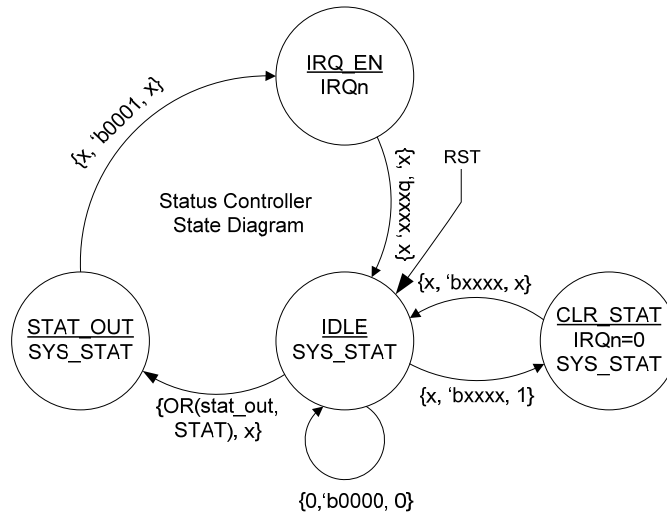
Figure 5: Example Interface Controller State Machine ASM diagram

3.3.1 Status Controller (SC):

The SC provides status of the interface and S_uMMIT to the LEON 3FT by capturing events related to interface and SuMMIT operations. The inputs that determine which state (or action) to process are based on the Interface Controller clear (clr), Interface Controller status bits (stat_out), and S_uMMIT status (YF_INTB, MSG_INTB, TERACTION, READYB) bits. The Example IRQ_EN State Flowchart in Figure 7 shows the status captured in the register and evaluation of the S_uMMIT status bits to determine if the IRQ_EN signal will be enabled. Additionally, the output of this controller can aid in troubleshooting problems in the development of the overall interface. A description of each signal is listed in Table 3.

Table 3: SC Signal Description Table

| Signal | Signal Direction | Purpose/Description |
|----------|------------------|--|
| RST | Input | Global interface reset |
| stat_out | Input | Interface Status signal to indicate the status of the task being performed |
| STAT | Input | S _u MMIT YF_INB, MSG_INTB, TERACTION, READYB status signals |
| clr | Input | Clear signal from the IS to clear the output of the SC |
| SYS_STAT | Output | Concatenation of stat_out and STAT held on the output to the UT699 |
| IRQn | Output | Single output for triggering interrupts in the UT699 |



Inputs: {stat_out, STAT(YF_INTB, MSG_INTB, TERACTION, READYB), clr}
 Outputs: {SYS_STAT(stat_out, STAT), IRQn}

Figure 6: Example Status Controller State Machine ASM diagram

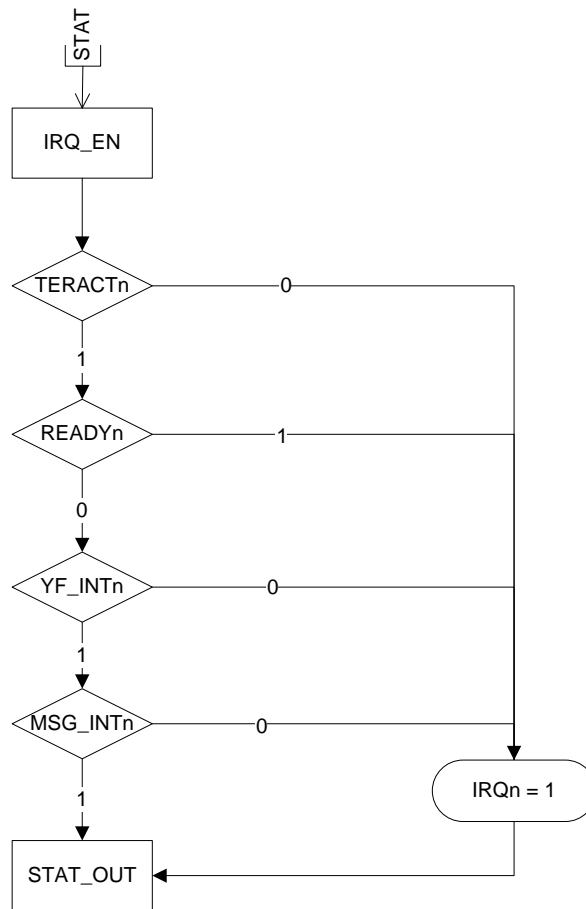


Figure 7: Example IRQ_EN State Flowchart

3.3.2 DMA Controller (DMAC):

The DMAC monitors requests from the SμMMIT and LEON 3FT for control of the shared bus for access to the SμMMIT registers and shared memory, determines priority for each request, provides control signals to the Interface Data Path Controller and status of current actions. The LDMACKn signal from the UT699 indicates processing of UT699 data on the share bus. The LDTACKn signal to the UT699 indicates the data transferred is complete through the IC. A description of each signal is listed in Table 4.

Table 4: DMAC Signal Description Table

| Signal | Signal Direction | Purpose/Description |
|----------|------------------|--|
| RST | Input | Global interface reset |
| RCSn | Input | SμMMIT Ram Chip Select for timing control of dtackb |
| dmackn | Input | UT699 LDMACKn or SμMMIT DMACKn |
| dmarn | Input | UT699 LDMARn or SμMMIT DMARn |
| dmagb | Output | DMA Grant signals to UT699 (LDMAGn) or SμMMIT (DMAGn) |
| stat_out | Output | DMAC Status Output signal to indicate the status of the task being performed |
| dtackb | Output | DTACK signal to UT699 (LDTACKn) or SμMMIT (DTACKn) |

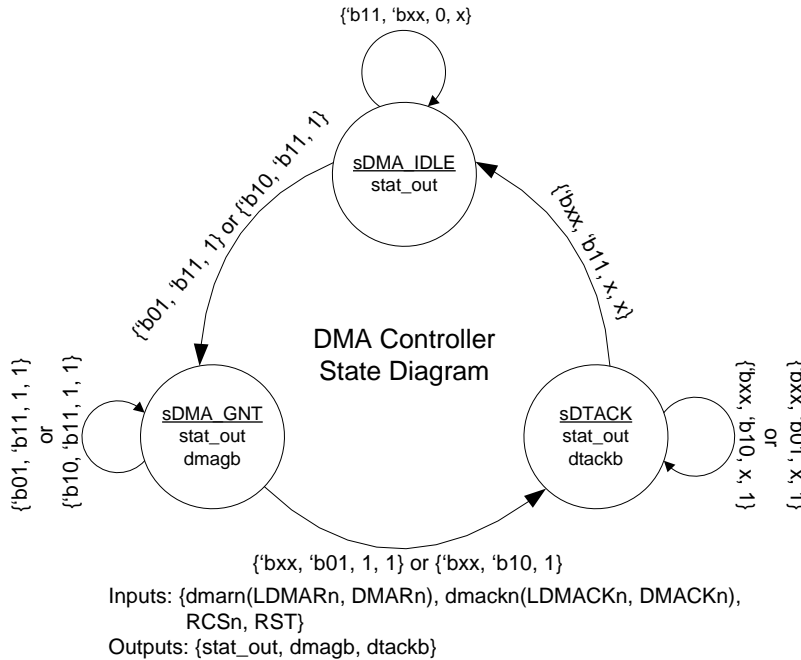


Figure 8: Example DMA Controller State Machine ASM diagram

3.3.3 Data Path Controller (DPC):

The DPC monitors control signals from the DMA Controller and Interface Controller to determine how to direct data and address flow. This controller represents the physical connections and logic required to control, address, and data I/O to the shared memory or SμMMIT (Figure 9). This controller has logic for controlling data path buffers (Figure 10) for address, data, read/write, and select. Additionally it controls the direction for data. Each logic path ends on the output of a control signal and only changes when the inputs change. A description of each signal is listed in Table 5.

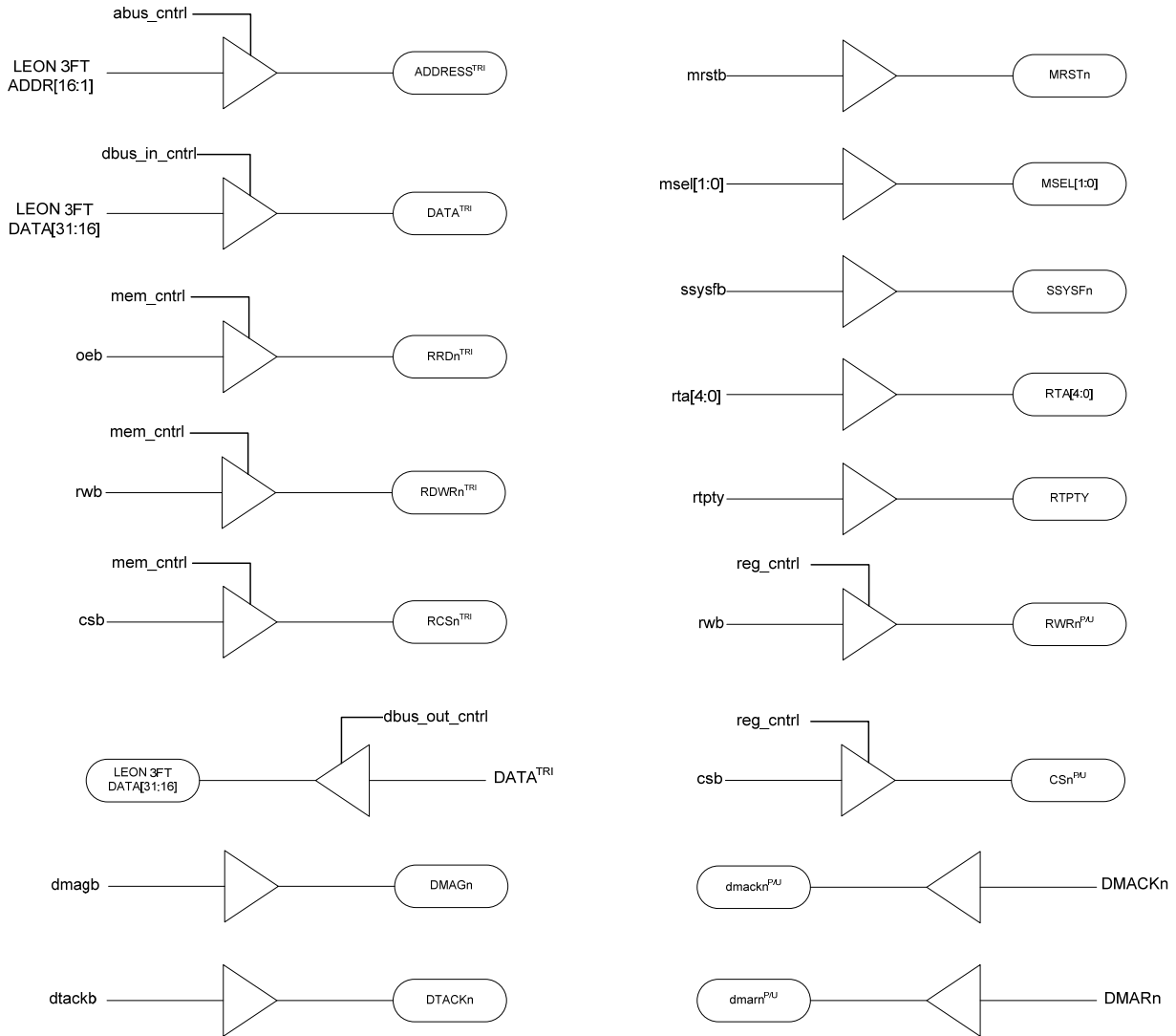


Figure 10: Example Data Path Controller Buffer Diagram

3.4 Software Requirements:

Modern designs balance the use of hardware with the versatility of software. The UT699 relies on the versatility of software to control internal and external devices. Controlling the interface requires software drivers written for the UT699 to control the interface for a given set of tasks. A project may require a subset or all of the blocks in Figure 11, depending on the software requirements. As a minimum, the driver should include interrupt handling, read and write to the interface, read and write to the shared memory, read and write to the SμMMIT registers, device startup, and device shutdown functionality. A Real Time Operating System (RTOS) may require the use of additional blocks. The following is a function description of each block:

- Interrupt Block:
 - Register the Interrupt Service Routines (ISR) related to operating the device driver
 - ISR for Interface Access
 - ISR for DMA control
 - ISR for device failures
 - Enable the ISR
 - Disable the ISR
- Device Startup:
 - Places SμMMIT in a power on state
 - Configures Memory Controller, GPIO, and Interrupt Controller registers for operations
 - Allocates memory for driver operations

- Configures shared memory for 1553 operations
 - Configures SμMMIT registers for operations
- Device Install:
 - Permits other software to install driver
 - Installs and register device parameters in a RTOS
- Device Enable:
 - Allows software to enable use of device using pre-allocated of memory
- Write Interface:
 - Translates and supplies address' associated with Interface internal registers
 - Controls operations to write Data to Interface internal registers
 - Waits for ISR to notify to continue to write to registers
- Read Interface:
 - Controls operations associated with Interface internal registers
 - Store data
 - Waits for ISR to notify to continue to read to registers
- Read Shared Memory:
 - Send DMA request
 - Wait for ISR to notify program when DMA grant is received
 - Send DMA acknowledge
 - Controls operations associated with shared memory access
 - Store data
 - Waits for ISR to notify to continue to shared memory reads
- Write Shared Memory:
 - Send DMA request
 - Wait for ISR to notify program when DMA grant is received
 - Send DMA acknowledge
 - Controls operations associated with shared memory access
 - Waits for ISR to notify to continue to write to shared memory
- Write SμMMIT:
 - Send DMA request
 - Wait for ISR to notify program when DMA grant is received
 - Send DMA acknowledge
 - Controls operations associated with SμMMIT register access
 - Waits for ISR to notify to continue to write to the SμMMIT
- Read SμMMIT:
 - Send DMA request
 - Wait for ISR to notify program when DMA grant is received
 - Send DMA acknowledge
 - Controls operations associated with SμMMIT register access
 - Store data
 - Waits for ISR to notify to continue to read to the SμMMIT
- Device Acquire:
 - Confirms SμMMIT ready
 - Permits other interfaces to “Lock” control of Interface driver
- Device Release:
 - Indicates SμMMIT idle
 - Permits other interfaces to release the “Lock” to control the Interface driver
- Device Uninstall:
 - Permits other software to remove driver
 - Un-configures Memory Controller, GPIO, and Interrupt Controller registers for operations
 - Release memory allocated for driver operations
- Device Disable:
 - Allows software to disable use of device without reallocation of memory
- Device Shutdown:
 - Places SμMMIT in a power down or sleep state

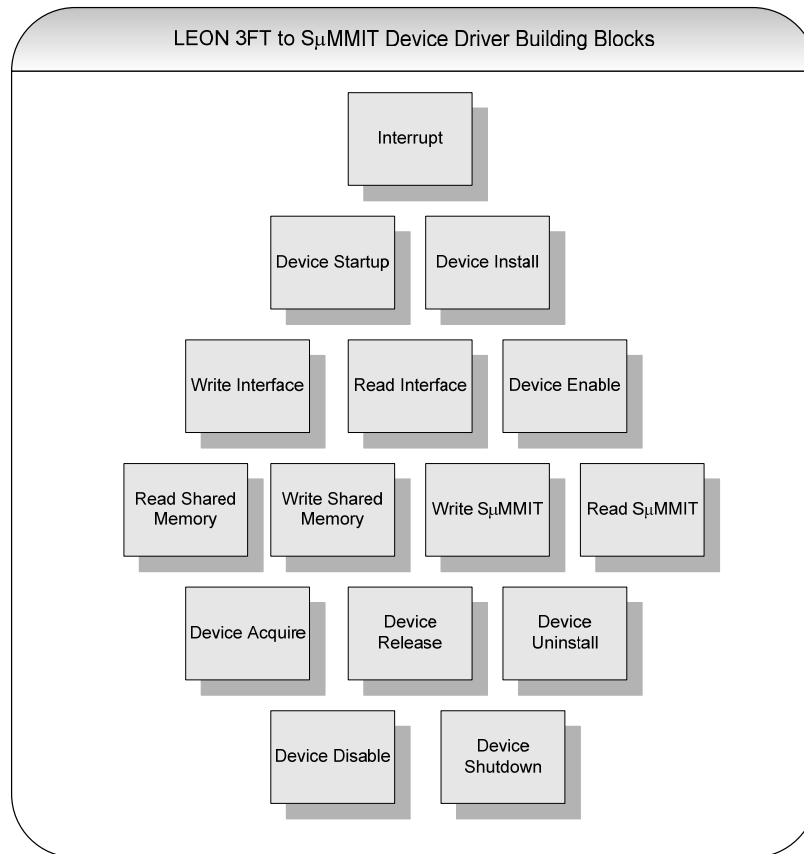


Figure 11: Example LEON3 FT Software Driver Flowchart

4.0 Summary and Conclusion:

The basic requirements of interfacing the LEON 3FT to the S μ MMIT are:

- A FPGA that meets design parameters and provides both 3V and 5V selectable I/O
- Synthesized RTL of the Interface Controller
- A LEON 3FT driver for the Interface (Access, Control, Feedback)

Design of the Interface Controller must provide correct timing and control to the LEON 3FT and the S μ MMIT to have a successful system.

Considerations for development of software drivers:

- Standalone vs. OS
- Required functionality
- Ease of use (Interface and Documentation)
- Intersystem operability

The UT699 shows an increased capability by utilizing the memory mapped I/O controls and General Purpose I/O (GPIO) to control many different devices.

5.0 References:

- Aeroflex Colorado Springs Inc., The Enhanced S μ MMIT Family Product Handbook, Oct. 1999
- Aeroflex Colorado Springs Inc., UT699 LEON 3FT/SPARC™ V8 MicroProcessor Advanced User Manual, Aug. 2010
- Aeroflex Colorado Springs Inc., UT699 32-bit Fault-Tolerant SPARC™ V8/LEON 3FT Processor Data Sheet, Feb. 2011