DPRAM Interface

Uros Minoski

April 2024

1 Introduction

This document aims to elucidate the functionalities of the 7028 - 64k x 16 Dual-Port RAM and its interface with the ISA bus.

2 7028 Dual-Port RAM

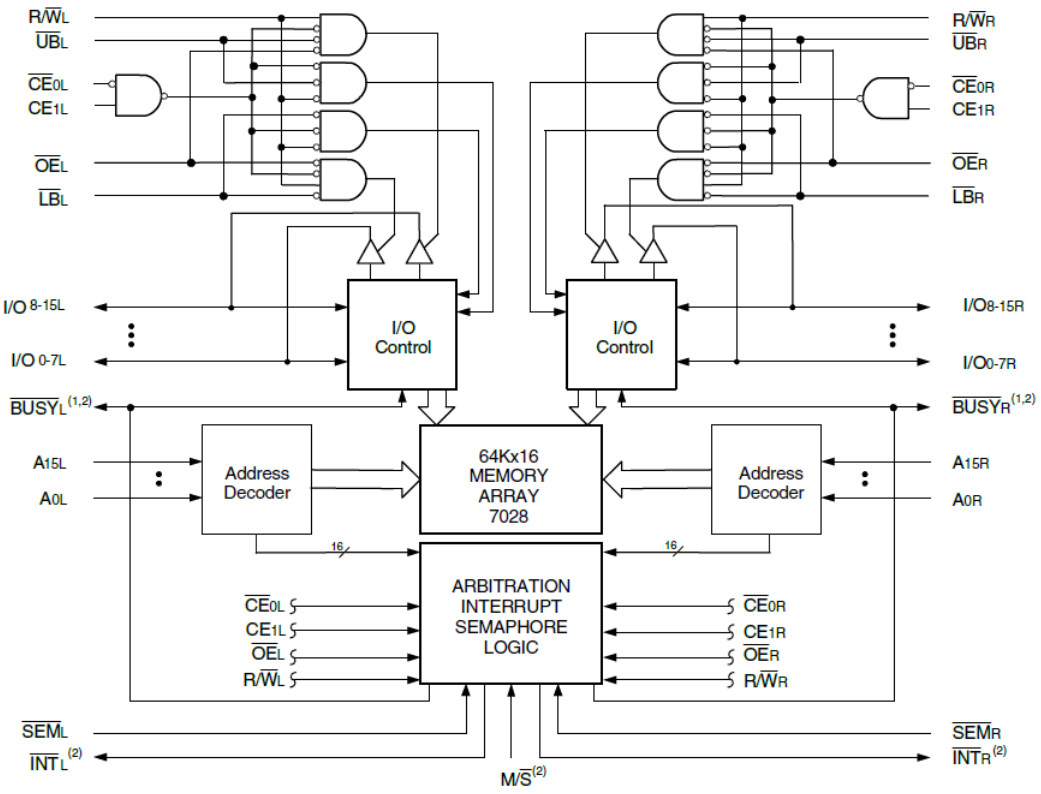

Figure 1: Functional block diagram of 7028 - 64k x 16 Dual-Port RAM

The IDT7028 is a high-speed 64K x 16 Dual-Port Static RAM. The IDT7028 is designed to be used as a stand-alone 1024K-bit Dual-Port RAM or as a combination MASTER/SLAVE Dual-Port RAM for 32-bit-ormore word systems. Using the IDT MASTER/SLAVE Dual-Port RAM approach in 32-bit or wider memory system applications results in fullspeed, error-free operation without the need for additional discrete logic. This device provides two independent ports with separate control, address, and I/O pins that permit independent, asynchronous access for reads or writes to any location in memory. An automatic power down feature controlled by the chip enables ($CE0$ and $\overline{CE1}$) permit the on-chip circuitry of each port to enter a very low standby power mode. Fabricated using CMOS high-performance technology, these devices typically operate on only 1W of power. The IDT7028 is packaged in a 100-pin Thin Quad Flatpack (TQFP) (Fig. 2).

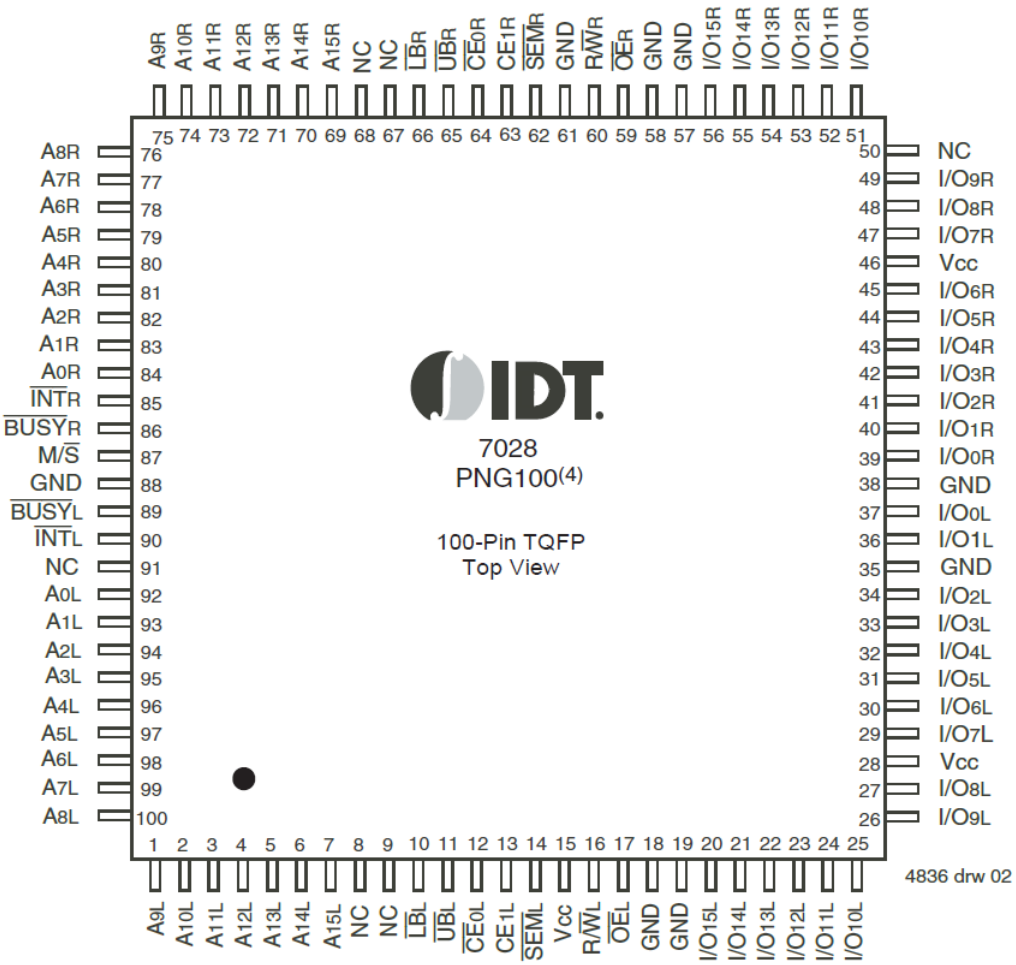

Figure 2: Pin configuration of IDT7028.

Mailbox address space is 0000_h - $FFFF_h$. Last two address locations FFFEh and FFFFh are reserved for writing/reading interrupt messages. Accessing mailbox $\overline{SEM}$ must be inactive and $\overline{CE0}$ and $CE1$ must be active. Access to the semaphores has its own address space, which is 0000_h - 0007_h and $\overline{SEM}$ must be active and $\overline{CE0}$ and $CE1$ must be inactive.

Interrupts

Left port interrupt flag ($\overline{INT}_L$) is asserted when right port writes to the address location $FFFE_h$. The left port clears the flag by reading form that address location ($\overline{INT}_L$). Likewise, right port flag ($\overline{INT}_R$) is asserted when left ports writes to the address location $FFFF_h$. The right port clears the flag by reading from that address location ($FFFF_h$). The programmer has full discretion to write whatever they please to these address locations. Interupt lines must be enabled.

Busy Logic

Busy logic provides hardware indications that both ports of the RAM have accessed the same address location at the same time. It also allow one of the access to proceed, an the other is notified by $\overline{BUSY}$ line. Write signal is gated internally, with $\overline{BUSY}$ line, to prevent simultaneous writing to the same address location.

In master mode ($M/\overline{S} = 1$) $\overline{BUSY}$ is an output, in slave mode is an input. If there is no need for busy logic, chip could be set as slave and $\overline{BUSY}$ must be high for enabling access, and low for disabling access. Master/Slave select is used in width and/or depth RAM expansion (placing more then one Dual-Port RAM chips).

Semaphores

Semaphores are realised as hardware components (Fig. 3). IDT7028 does not use its semaphore flags to control any resource through hardware, thus, allowing the system designer total flexibility in system architecture.

Semaphores provide hardware assist for a use assignment method called "Token Passing Allocation". In this method, state of semaphore latch is used as a token to indicate that shared resource is in use.

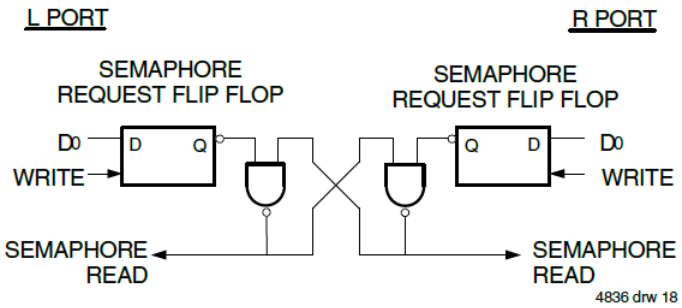

Figure 3: IDT7028 semaphore.

The semaphore flags are active **LOW**. A token is requested by writing a zero into a semaphore latch and is released when same side writes one to that latch.

The eight semaphore flags reside within the IDT7028 in a separate memory space from the Dual-Port RAM. That address space is accessed by placing a **LOW** input on the $\overline{SEM}$ (which acts as a chip select for semaphore flags) and using other control pins (Address, $\overline{CE}$ and $R/\overline{W}$) as they would be used in accessing a standard Static RAM. each of the flag has unique address which can be accessed by either side through address pins $A_0 ... A_2$. Other address pin have no effect. When writing to a semaphore, only D_0 is used. While reading all data pins are used and $\overline{OE}$ (Output Enable) must be **LOW**. **Semaphore flag is spread into all data bits** (in read), so that if flag is one, read value on data bus will be all ones, if the flag is zero, read value will be all zeros.

If a **LOW** level is written into an unused semaphore location, that flag will be set to a zero. When a one is written to the same location from the same side, the flag will be set to one for both sides (unless a semaphore request from the other side is pending) and the can be written to by both sides. So if one side is able to write zero to the semaphore locks out writes from the other side. A zero written to the taken semaphore, will be stored in the semaphore request latch for that side until the semaphore is freed by the first side.

A sequence **WRITE/READ** must be used by the semaphore in order to guarantee that no system level contention will occur.

It is **important** to note that failed semaphore request must be followed by either repeated reads or by writing a one into the same location to clear zero pending from semaphore request latch.

The critical case of semaphore timing is when both sides request a single token by attempting to write a zero into it at the same time. If one side is earlier than the other in making the request, the first side to make the request will receive the token. If both requests arrive at the same time, the assignment will be arbitrarily made to one port or the other.

Initialization of the semaphores is not automatic and must be handled via the initialization program at power-up. Since any semaphore request flag which contains a zero must be reset to a one, all semaphores on both sides should have a one written into them at initialization from both sides to assure that they will be free when needed.

3 Dual-Port RAM Interface with the ISA Bus

Interface is made for decoding addresses and translating control signals from ISA bus to the Dual-Port RAM. Fig. 4 shows functional block diagram of interconnection of interface, Dual-Port RAM and two ISA devices.

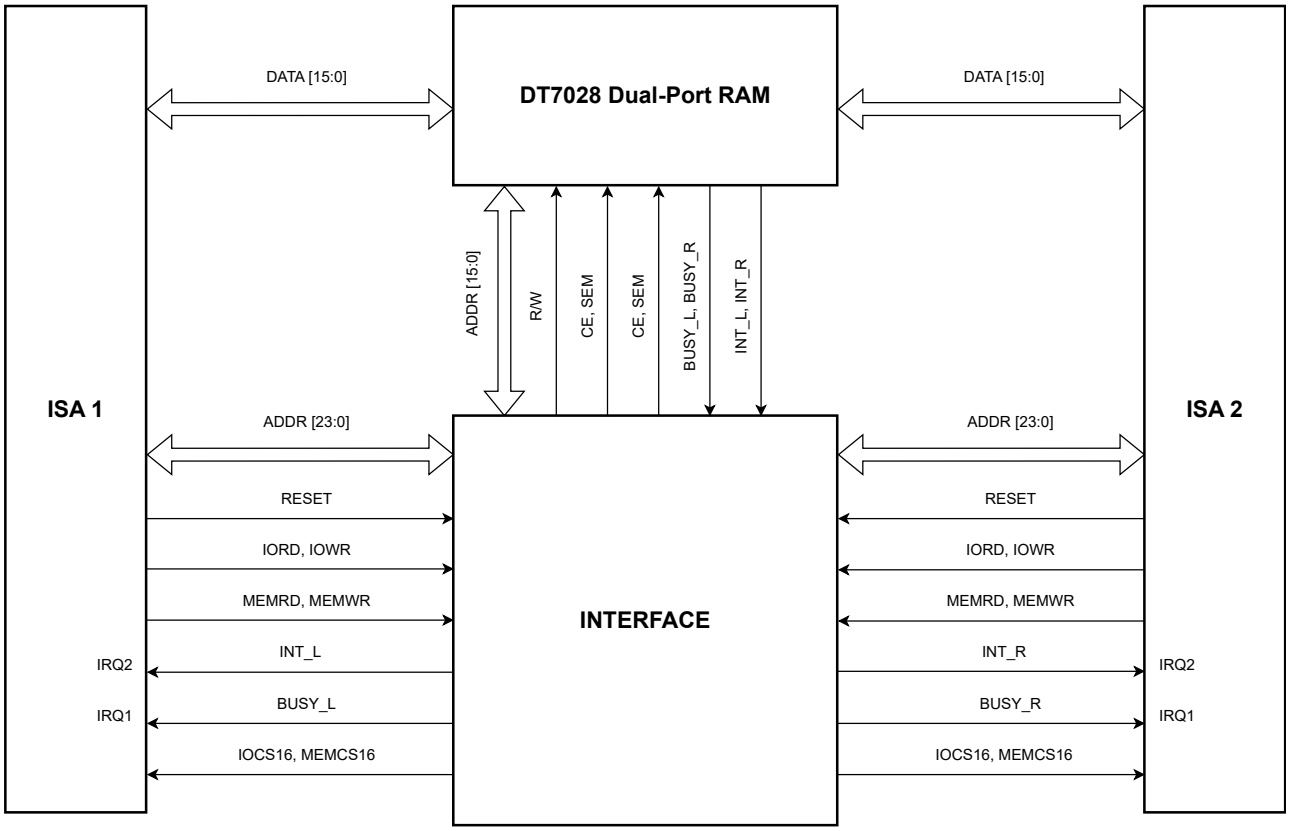

Figure 4: Functional block diagram of interface.

Random memory Access is addressed with addresses 000000_h - $00FFFF_h$ plus base address, and setting **LOW** $\overline{MEMRD}$ (for reading from RAM) or $\overline{MEMWR}$ (for writing to RAM). Last two address locations $FFFE_h$ and $FFFF_h$ are reserved for interrupts.

Semaphores are addressed with addresses 000000_h - 000007_h plus base address, and setting **LOW** $\overline{IORD}$ (for reading semaphore flag) or $\overline{IOWR}$ (for setting/realising semaphore flag).

By writing to address location reserved for interrupts, interrupt line will trigger interrupt on ISA bus. Reading from those locations will clear interrupt flags, as explained before.

In asynchronous regime, if both ports address same location in RAM, $\overline{BUSY}$ line will be set **LOW**, that is connected to interrupt pin on ISA bus. $\overline{BUSY}$ line will be **LOW** as long as other port doesn't finish bus cycle on RAM.