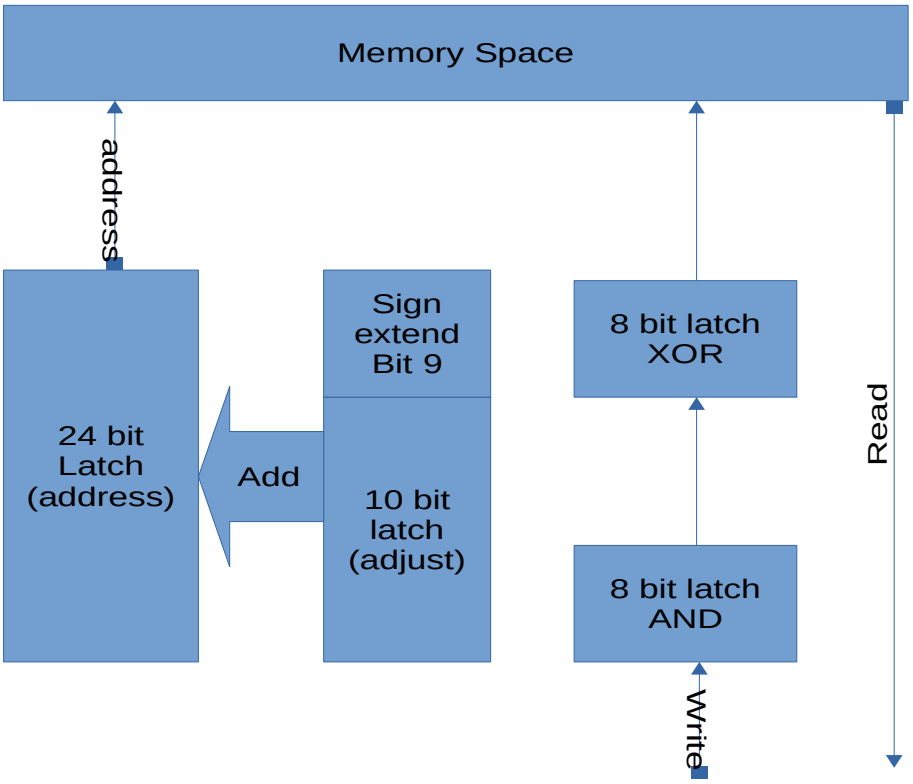
Software Driven Blitter/DMA

This is an outline of a software driven blitter for the F256 Junior.

The concept behind this is that for reasons of both timing and FPGA real-estate it is difficult to build a blitter or DMA that operates automatically.

However, a combination of the limitations of the 65C02 as a processor and the memory mapping of the F256 means it struggles to simply and efficiently process data in its 1M (?) address space.



Plan A

This is a single data access/update element. There should be two of these ; there are at least two use cases for this (1) copying data and (2) rapid drawing of lines,and circles that use slope algorithms.

Address

The address part consists of a 24 bit address latch with a 10 bit signed adjustment latch which can be added to that address. This needs to allow, as a bare minimum additions of 0,+/-319,+/-320 and +/-321.

The addition can either be done on every read/write, or not at all. This needs to be done without having to reset the adjustment latch.

The simplest way of doing this I believe is to have two read and two write ports for each pair of data access/update elements, one causes the adjust latch to be added, the other leaves it unchanged. The LSB of the port address could be used to gate the addition.

This allows sequential processing of data – e.g. read action write.by each element individually or in concert.

Data

When the Data is read through an I/O port the data read is that at the address in question.

When it is written there is the option to process that data as it goes in. This takes two phases.

1) It is firstly ANDed with a fixed value.

2) It is exclusive Ored with a fixed value.

Rationale

The AND/XOR option allows various types of writes to memory, which for bitmaps (mostly !) allows graphic images to be written in various ways other than strict copying. When drawing graphics to the bitmap it allows colour to be ANDed, ORed, XORed, or simply copied into the memory automatically, allowing the use of some palette effects.

This could of course be done purely manually but this could be considered as some extra functionality at low cost (I do not know how propagation would be affected here).

Port requirements

2 x 8 bit Read (add, don’t add)

2 x 8 bit Write (add, don’t add)

3 x 8 bit set address register

2 x 8/2 bits adjust address register

2 x AND/OR processing registers

All opinions welcomed.

Paul Robson 18/10/2022