Zolatron: Memory & Addressing

Last edited: 8 Jun 2025

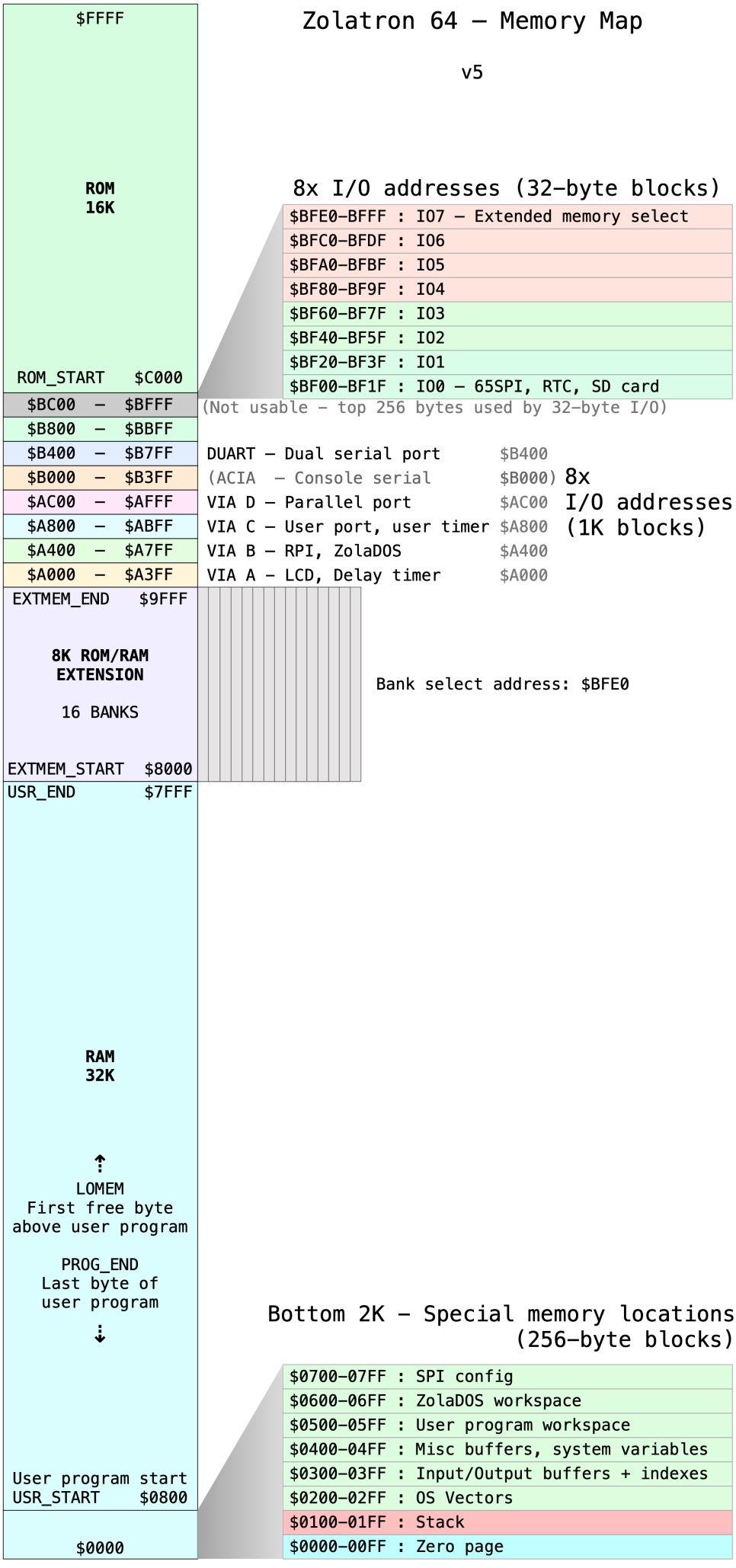
Main sections

	Address Range	Used for
	$0000 - $07FF	System variables, buffers etc.
	$0800 - $7FFF	User RAM
	$8000 - $9FFF	Extended ROM/RAM
	$A000 - $BFFF	I/O
	$C000 - $FFFF	System ROM

Lower 2K

Pages 0–7 ($0000–$07FF) are reserved for system use:

Pg	Address Range	Used for	Defined in:
0	$0000-$00FF	Zero page variables	cfg_page_0.asm
1	$0100-$01FF	6502 system stack	–
2	$0200-$02FF	OS indirection table	cfg_page_2.asm
3	$0300-$03FF	STDIN & STDOUT buffers and indices	cfg_main.asm
4	$0400-$04FF	Buffers etc - includes STR_BUF, variables for temporary values & maths operations	cfg_page_4.asm
5	$0500-$05FF	User program workspace	–
6	$0600-$06FF	ZolaDOS workspace	cfg_ZolaDOS.asm
7	$0700-$07FF	SPI workspace	cfg_page_7.asm


Zolatron 64 – Memory Map
v5
$FFFF
ROM
16K
ROM_START $C000
8x I/O addresses (32-byte blocks)
$BFE0-BFFF : IO7 – Extended memory select
$BFC0-BFDF : IO6
$BFA0-BFBF : IO5
$BF80-BF9F : IO4
$BF60-BF7F : IO3
$BF40-BF5F : IO2
$BF20-BF3F : IO1
$BF00-BF1F : IO0 – 65SPI, RTC, SD card
$BC00 – $BFFF (Not usable – top 256 bytes used by 32-byte I/O)
$B800 – $BBFF
$B400 – $B7FF DUART – Dual serial port $B400
$B000 – $B3FF (ACIA – Console serial $B000)
$AC00 – $AFFF VIA D – Parallel port $AC00
$A800 – $ABFF VIA C – User port, user timer $A800
$A400 – $A7FF VIA B – RPI, ZolaDOS $A400
$A000 – $A3FF VIA A – LCD, Delay timer $A000
8x I/O addresses (1K blocks)
EXTMEM_END $9FFF
8K ROM/RAM EXTENSION
16 BANKS
EXTMEM_START $8000
Bank select address: $BFE0
USR_END $7FFF
RAM
32K
LOMEM
First free byte above user program
PROG_END
Last byte of user program
User program start
USR_START $0800
$0000
Bottom 2K – Special memory locations (256-byte blocks)
$0700-07FF : SPI config
$0600-06FF : ZolaDOS workspace
$0500-05FF : User program workspace
$0400-04FF : Misc buffers, system variables
$0300-03FF : Input/Output buffers + indexes
$0200-02FF : OS Vectors
$0100-01FF : Stack
$0000-00FF : Zero page

ZERO PAGE

Defined in cfg_page_0.asm.

Address	Name	Description
$E0, $E1	MSG_VEC	Pointer to a message (to print)
$E2, $E3	FUNC_RES_L, FUNC_RES_H	To hold a 16-bit return value for a
$E4	FUNC_RESULT	Holds a 1-byte return value for a
$E5	FUNC_ERR	Stores an error code returned from a
$E6, $E7	TBL_VEC_L, TBL_VEC_H	LSB and MSB of a location within a lookup
$E8, $E9	TMP_ADDR_A_L, TMP_ADDR_A_H	Temporary 16-bit address (TMP_ADDR_A is an alias for TMP_ADDR_A_L)
$EA, $EB	TMP_ADDR_B_L, TMP_ADDR_B_H	Temporary 16-bit address (TMP_ADDR_B is an alias for TMP_ADDR_B_L)
$EC, $ED	TMP_ADDR_C_L, TMP_ADDR_C_H	Temporary 16-bit address (TMP_ADDR_C is an alias for TMP_ADDR_C_L)
$EE, $EF	FILE_ADDR	
$F0, $F1	PROG_END	Address of last byte of user program
$F2, $F3	LOMEM	First available byte after user prog
$F4	STDIN_STATUS_REG	STDIN flags
$F5	SYS_REG	System flags
$F6	IRQ_REG	IRQ flags
$F7-$FB	- not used -	5 bytes
$FC, $FD	USRINT_VEC	Indirect jump vector pointing to userland
$FE, $FF	USRINTTRN_VEC	Indirect jump vector returning from

PAGE 3 – $0300–$03FF

I/O buffers, temp storage etc

Defined in cfg_main.asm.

$0300 - $037E	STDIN_BUF	127
$037F	STDIN_IDX	1
$0380 - $03FE	STDOUT_BUF	127
$03FF	STDOUT_IDX	1