V6502WS general purpose 8 bit microprocessor datasheet

Design by: Valerio Venturi (valerioventuri@gmail.com)

The V6502WS is a VHDL RTL softcore 100% software compatible with the original silicon 65C02 CPU but with some improvements:

1) an 16 bit S stack pointer register instead the original 8 bit S register (at RESET the S is automatically initialized to $01XX in order to preserve 6502 compatibility).
2) an Z zeropage 8 bit relocation register (at RESET this register is set to zero).
   The Z register is used to form the MSB address (bits 15 to 8) when any zero page instruction is executed:

   original 6502/65C02 zeropage addressing:

bit 15 … 8	7 … 0
forced to $00	$XX (ZP opcode offset)

   V6502WS:

bit 15 … 8	7 … 0
Z register value	$XX (ZP opcode offset)

   The Z register can be useful in a multithreading system where each thread needs to have a private zeropage area or to access to I/O peripherals more quickly.
   (Unlike 65C02 the RES input cannot be used as additional interrupt request signal because the contents of S and Z registers are losses).
3) TAZ and TZA instructions to load and save Z register
4) PHR and PLR instructions to push and pull simultaneously A,X,Y registers on stack, they can be used instead the classic 65C02 sequence used to save and restore registers:

```
PHA          ;use only PHR to push A,X,Y to stack in the same sequence (four clock cycles instead six)
PHX
PHY
…...
…...
PLY          ;use only PLR to pull Y,X,A in the same sequence (five clock cycles instead nine)
PLX
PLA
```

   These instructions can be used to speed up an interrupt routine if it uses all registers, for example in a multithreading context switch, or in a subroutine that uses all registers.
   The major advantages are: speed up the code (nine clock cycles instead fifteen) and save memory space (two byte instead six to obtain the same function).
   Also the PHR and PLR instructions don't affect any flags on P status register, this may be useful if a subroutine returns something on P register.
5) ISP and TSP are atomic instructions to load and save the entire 16 bit S stack pointer, the old TXS,TSX instructions keep their original function but only the LSB portion of stack pointer is affected.
6) XYX,XAX,XAY instructions to exchange between registers.
   These instructions don't affect any flags on P status register.
7) Interrupt and RTI instruction save and restore automatically the Z register, so after an interrupt is recognized the Z register is saved on stack and automatically cleared to zero in order to point the original 6502 zeropage
8) Improved execution time for some instructions and addressing modes.
9) Like the original 65C02 all unused opcodes are treated as NOPs.

The V6502WS has a compact architecture design because all instructions are microcoded in a PLA structure, therefore the CPU is scalable and well suitable to put it on FPGA, CPLD and also ASICs.
For example in order to save space on FPGA, if some opcodes or addressing modes are not used on your project, they may be removed from microcode PLA simply by put a comment to related microcode lines.

New Opcodes:

```
PHR (0x8B)   push AXY to stack   (flags: unaffected)
PLR (0xAB)   pull AXY from stack (flags: unaffected)
TXY (0x9B)   transfer X to Y  (flags: NVZ)
TYX (0xBB)   transfer Y to X  (flags: NVZ)
XYX (0xEB)   exchange X and Y (flags: unaffected)
XAX (0x0B)   exchange A and X (flags: unaffected)
XAY (0x2B)   exchange A and Y (flags: unaffected)
ISP (0x4B)   copies X to S (lsb) and A to S (msb) (flags: unaffected)
TSP (0x5B)   copies S (lsb) to X and S (msb) to A (flags: unaffected)
TAZ (0x1B)   transfer A to Z  (flags: NVZ)
TZA (0x3B)   transfer Z to A  (flags: NVZ)
JSR (0xFC)   JSR (ABSOLUTE_INDIRECT,X) jump to subroutine (flags: unaffected)
```

16 bit stack pointer initialization example:

```
     lda    #$1f        ;msb stack pointer
     ldx    #$ff        ;lsb stack pointer
     isp                ;set top of stack at 0x1FFF
```

NOTE: to preserve compatibility the S msb register is set automatically to 0x01 after reset

zpage initialization example:

```
     lda    #$c0
     taz                ;set zero page at 0xC000
```

NOTE: to preserve compatibility the Z register is set automatically to zero after reset

jump to subroutine from address table

```
     ldx    #2              ;select subroutine2
     jsr    (subr_table,x)  ;call subroutine2

     subr_table   .word subroutine1, subroutine2, subroutine3, subroutine4
```

FPGA implementation

The project was tested for many months on ALTERA Terasic DE2-115 board on Cyclone IV and it uses 1693 LCs. The V6502WS running perfectly @25 MHZ with VGA 640X480 32K colors and a UART with preemptive multitasking operating system.