The following batch files are transferred to the Desktop from the GitHub location "c:\Users\vidal\Documents\GitHub\jpeg-2000-test". These 3 batch files are used to upload a hex file to the XulA2-LX9 SDRam, download from XulA2-LX9 SDRam to a file, and compare the values of the upload/download files.

wr.bat

```
cd "c:\Program Files (x86)\XSTOOLs"
xsload.exe -usb 0 -f hex -ram "c:\Users\vidal\My Documents\GitHub\jpeg-2000-test\ipython_fixbv\lena.hex"
```

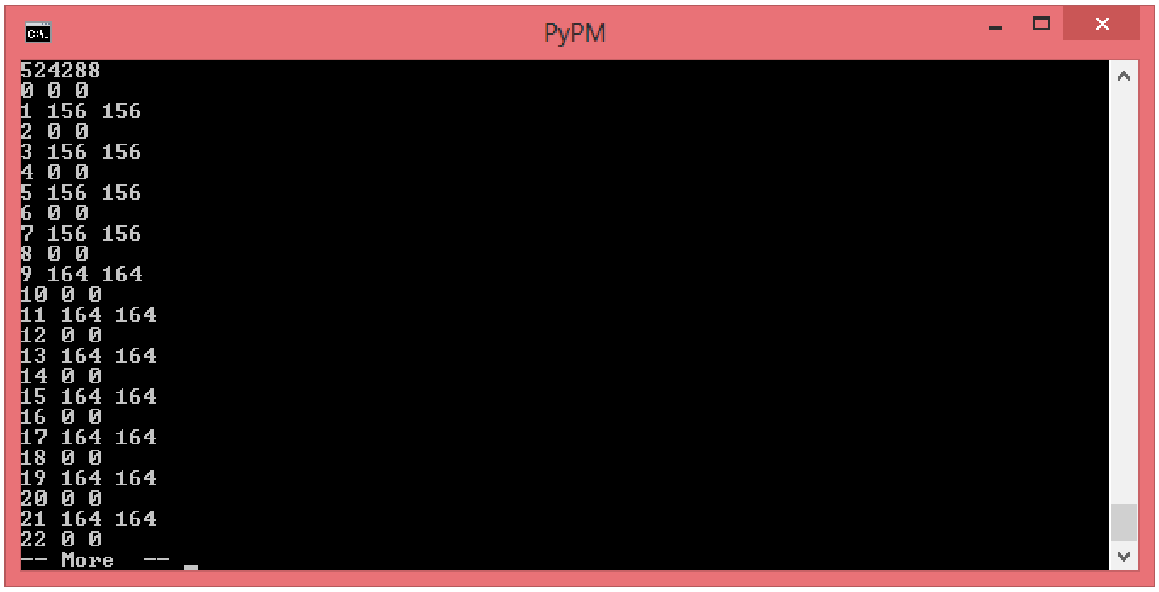
rd.bat

```
m:
dir ttt.hex
del ttt.hex
"c:\Program Files (x86)\XSTOOLs\xsload.exe" -usb 0 -f hex -u 0x00000 0x7FFFF -ram ttt.hex
dir ttt.hex
copy ttt.hex "c:\Users\vidal\Documents\GitHub\jpeg-2000-test\ipython_fixbv"
dir  "c:\Users\vidal\Documents\GitHub\jpeg-2000-test\ipython_fixbv\ttt.hex"
pause
```

cmp.bat

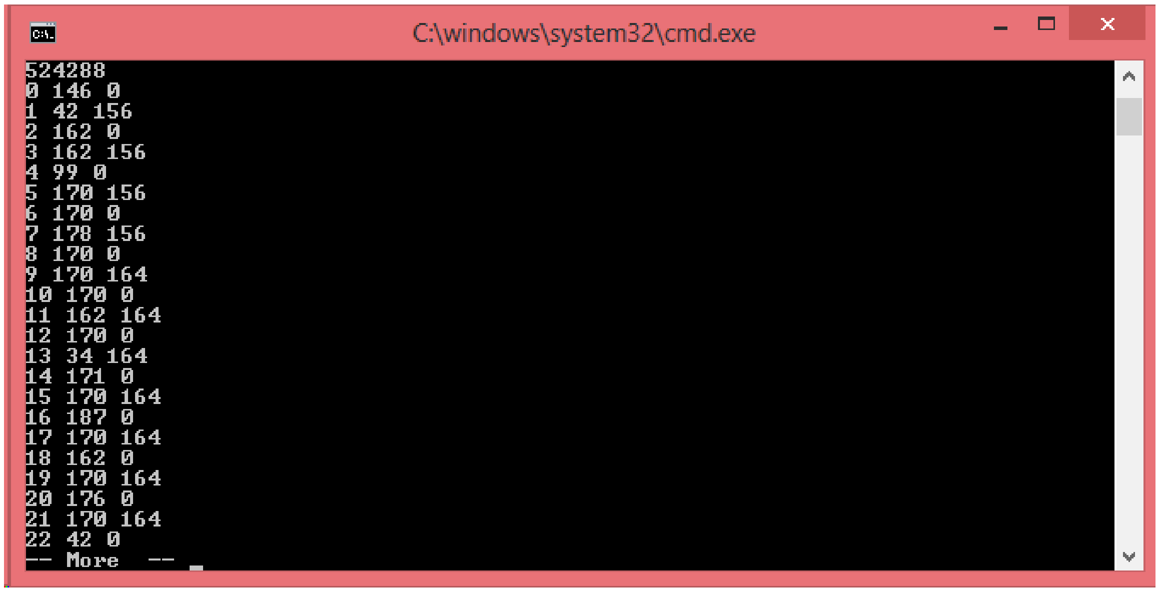
```
cd  "c:\Users\vidal\Documents\GitHub\jpeg-2000-test\ipython_fixbv"
python rd_lx9.py | more
```

The image below is showing the first 11 values of 524288 that are stored in the SDRam on XulA2-LX9. The number stored at address 0 & 1 is 00 9C hex or 0 156 decimal.



```
524288
0 0 0
1 156 156
2 0 0
3 156 156
4 0 0
5 156 156
6 0 0
7 156 156
8 0 0
9 164 164
10 0 0
11 164 164
12 0 0
13 164 164
14 0 0
15 164 164
16 0 0
17 164 164
18 0 0
19 164 164
20 0 0
21 164 164
22 0 0
-- More  --
```

If the SDRam is not written with lena.hex. The values between the file lena.hex & ttt.hex do not match as is the case in the image below.


C:\windows\system32\cmd.exe

```
524288
0 146 0
1 42 156
2 162 0
3 162 156
4 99 0
5 170 156
6 170 0
7 178 156
8 170 0
9 170 164
10 170 0
11 162 164
12 170 0
13 34 164
14 171 0
15 170 164
16 187 0
17 170 164
18 162 0
19 170 164
20 176 0
21 170 164
22 42 0
-- More  --
```

Information on how the file lena.hex was created.

Step 1 python lena2short.py

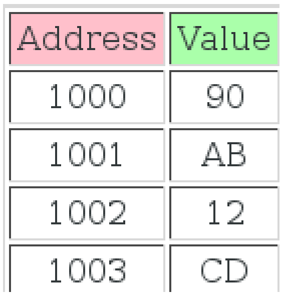
Step 2 python /bin/bin2hex.py tmp.bin lena.hex

So, the 4 bytes are: 90, AB, 12, CD where each byte requires 2 hex digits.

It turns out there are two ways to store this in memory.

Big Endian

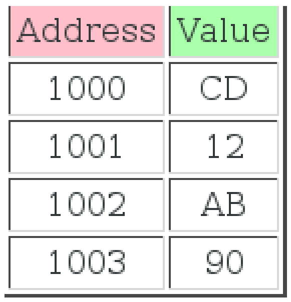
In big endian, you store the most significant byte in the smallest address. Here's how it would look:



Address	Value
1000	90
1001	AB
1002	12
1003	CD

Little Endian

In little endian, you store the *least* significant byte in the smallest address. Here's how it would look:



Address	Value
1000	CD
1001	12
1002	AB
1003	90

big endian view lena.hex

```
:020000040000FA

:10000000009C009C009C009C00A400A400A400A4F0
:1000100000A400A400A400A4009C009C00A400A4D0

                                  .
:10FFF000005C005C005C005C00640064006C006CF1

:00000001FF
```

little endian lena.hex.orig

```
:020000040000FA

:100000009C009C009C009C00A400A400A400A400F0

:10001000A400A400A400A4009C009C00A400A400D0

                                  .
:10FFF0005C005C005C005C00640064006C006C00F1

:00000001FF
```