In [1]:

```
import sys
sys.path.append('/home/xilinx')

# Needed to run inference on TCU
import time
import numpy as np
import pynq
from pynq import Overlay
from tcu_pynq.driver import Driver
from tcu_pynq.architecture import pynqz1

# Needed for unpacking and displaying image data
%matplotlib inline
import matplotlib.pyplot as plt
import pickle
```

In [2]:

```
overlay = Overlay('/home/xilinx/tensil_pynqz1.bit')
tcu = Driver(pynqz1, overlay.axi_dma_0)
```

```
Layout(64, 16, 24, 16)
allocating buffer of dtype <class 'numpy.uint64'> of size 7348224
{}
allocated Slice(8192, 2097152)
{8192: Slice(8192, 2097152)}
allocated Slice(2113536, 2097152)
DRAM0: length = 2097152, offset = 0x1691
DRAM1: length = 2097152, offset = 0x1792
wrote addr=8388600 size=8
wrote addr=8388592 size=8
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152)}
allocated Slice(4210688, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576)}
allocated Slice(5259264, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576), 5259264: Slice(5259264, 1048576)}
allocated Slice(0, 6)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576), 5259264: Slice(5259264, 1048576), 0: Slice(0, 6)}
freeing Slice(4210688, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 5259264: Slice(5259264, 1
048576), 0: Slice(0, 6)}
freeing Slice(5259264, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 0: Slice(0, 6)}
freeing Slice(0, 6)
```

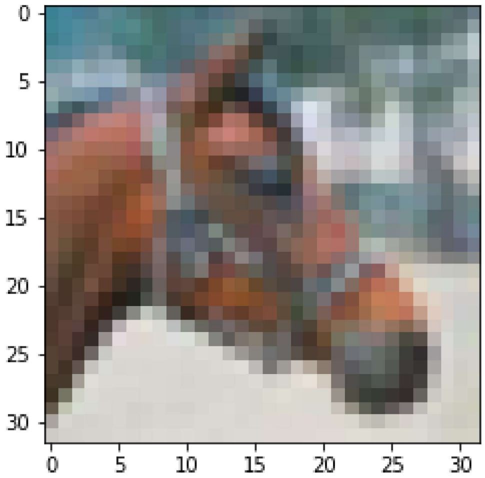
In [3]:

```
def unpickle(file):
    with open(file, 'rb') as fo:
        d = pickle.load(fo, encoding='bytes')
    return d

cifar = unpickle('/home/xilinx/cifar-10-batches-py/test_batch')
data = cifar[b'data']
labels = cifar[b'labels']

data = data[10:20]
```

```
labels = labels[10:20]

data_norm = data.astype('float32') / 255
data_mean = np.mean(data_norm, axis=0)
data_norm -= data_mean

cifar_meta = unpickle('/home/xilinx/cifar-10-batches-py/batches.meta')
label_names = [b.decode() for b in cifar_meta[b'label_names']]

def show_img(data, n):
    plt.imshow(np.transpose(data[n].reshape((3, 32, 32)), axes=[1, 2, 0]))

def get_img(data, n):
    img = np.transpose(data_norm[n].reshape((3, 32, 32)), axes=[1, 2, 0])
    img = np.pad(img, [(0, 0), (0, 0), (0, tcu.arch.array_size - 3)], 'constant', const
    return img.reshape((-1, tcu.arch.array_size))

def get_label(labels, label_names, n):
    label_idx = labels[n]
    name = label_names[label_idx]
    return (label_idx, name)
```

In [4]:

```
n = 7
img = get_img(data, n)
label_idx, label = get_label(labels, label_names, n)
show_img(data, n)
```



In [6]:

```
tcu.load_model('/home/xilinx/resnet20v2_cifar_onnx_pynqz1.tmodel')
```

```
wrote addr=0 size=570728
```

In [7]:

```
inputs = {'x:0': img}

start = time.time()
outputs = tcu.run(inputs)
end = time.time()
print("Ran inference in {:.4}s".format(end - start))
print()

classes = outputs['Identity:0'][:10]
result_idx = np.argmax(classes)
```

```
result = label_names[result_idx]
print("Output activations:")
print(classes)
print()
print("Result: {} (idx = {})".format(result, result_idx))
print("Actual: {} (idx = {})".format(label, label_idx))
```

```
wrote addr=0 size=8192
wrote inputs    0.00407s        0.00407s
wrote addr=8388600 size=8
wrote addr=8388592 size=8
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152)}
allocated Slice(4210688, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576)}
allocated Slice(5259264, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576), 5259264: Slice(5259264, 1048576)}
allocated Slice(6307840, 237192)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 4210688: Slice(4210688, 1
048576), 5259264: Slice(5259264, 1048576), 6307840: Slice(6307840, 237192)}
freeing Slice(4210688, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 5259264: Slice(5259264, 1
048576), 6307840: Slice(6307840, 237192)}
freeing Slice(5259264, 1048576)
{8192: Slice(8192, 2097152), 2113536: Slice(2113536, 2097152), 6307840: Slice(6307840, 2
37192)}
freeing Slice(6307840, 237192)
wrote program   0.133s  0.137s
read addr=256 size=16
read outputs    0.00882s        0.145s
Ran inference in 0.1478s

Output activations:
[-19.234375    -12.2890625   -8.1015625   -6.171875    -6.37890625
  -5.14453125  -7.47265625   1.734375    -9.5625      -7.7734375 ]

Result: horse (idx = 7)
Actual: horse (idx = 7)
```

In [ ]: