GF(2) Operations

December 16, 2019

```
In [5]: import numpy as np
```

```
In [28]: def xor(a, b):
             return np.logical_xor(a, b, dtype='uint8').astype("uint8")

         def and_(a, b):
             return np.logical_and(a, b, dtype='uint8').astype("uint8")


         def strip_zeros(a):
             return np.trim_zeros(a, trim='b')
```

```
In [3]: def gf2_add(a, b):
            a, b = strip_zeros(a), strip_zeros(b)

            N = len(a)
            D = len(b)
            F = min(N,D)

            if N == D:
                res = xor(a, b)

            elif N>D:
                res = np.concatenate((xor(a[:F], b), a[F:]))
            else:
                res = np.concatenate((xor(b[:F], a), b[F:]))

            return strip_zeros(res)
```

```
In [8]: a = np.array([1,0,1], dtype="uint8")
        b = np.array([1,1], dtype="uint8")
        gf2_add(a,b)
```

```
Out[8]: array([0, 1, 1], dtype=uint8)
```

```
In [14]: def gf2_mul(a, b):
             m=np.convolve(a,b)
             return strip_zeros(np.mod(m,2))
```

```
In [15]: a = np.array([1,0,1], dtype="uint8")
         b = np.array([1,1,1], dtype="uint8")
         gf2_mul(a,b)
```

```
Out[15]: array([1, 1, 0, 1, 1], dtype=uint8)
```

```
In [55]: def gf2_div(a, b):
             a, b = strip_zeros(a), strip_zeros(b)

             if not b.any():
                 return "error"
             elif len(b) > len(a):
                 q = np.array([])
                 return q, a

             else:
                 u = a.astype("uint8")
                 v = b.astype("uint8")

                 m = len(u) - 1
                 n = len(v) - 1
                 q = np.zeros((max(m - n + 1, 1),), "uint8")
                 r = u.astype("uint8")

                 for k in range(0, m - n + 1):
                     d = r[m - k].astype("uint8")
                     q[-1 - k] = d
                     r[m - k - n:m - k + 1] = xor(r[m - k - n:m - k + 1], and_(d, v))

                 r = strip_zeros(r)

             return q, r
```

```
In [60]: x = np.array([0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1], dtype="uint8")
         y = np.array([1, 0, 0, 0, 1], dtype="uint8")

         a,b = gf2_div(x, y)
         gf2_add(gf2_mul(a,y),b) == x
```

```
Out[60]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True])
```