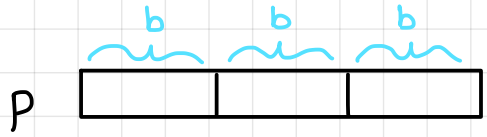
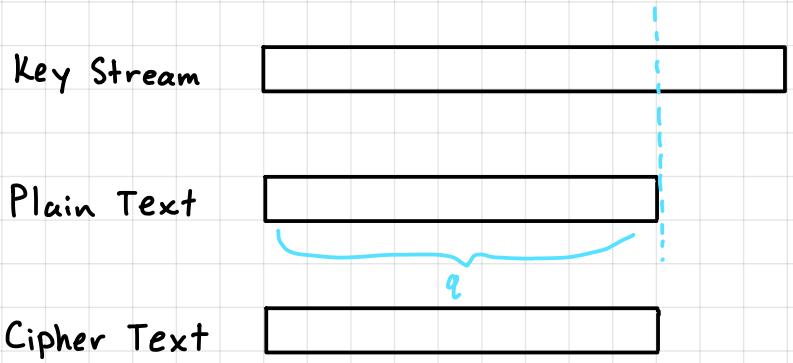
Padding



* Stream Cipher: no padding *



CBC / ECB

P → Padding → p' → Encrypt → C

C → Decrypt → p' → unpad → P

Need:

- unpad is inverse of pad
- p' is a multiple of b
- efficient

NOTE: There will always be padding for CBC and ECB in this class

10^* padding

- Append enough $\emptyset$'s to next multiple of b

b = 16 bits

p = 1111 0000 1111

p' = 1111 0000 1111 1000

---

p' = 1111 0000 1111 0000 1000 0000 0000 0000

padding

Sometimes there will be no trailing $\emptyset$'s

p = 1111 1111 1111 111

p' = 1111 1111 1111 1111 ← no $\emptyset$'s

p = 1111 1111 1111 111 ← strip only the last '1', no $\emptyset$'s

Mode Examples

Given:

$E: \{0,1\}^6 \rightarrow \{0,1\}^6$

$E(x) = ROTL(x, 2)$

If needed:

nonce = 101

IV = 110111

Counter start at <1>

10^* padding

Encrypt 0000 1111 0000 1111

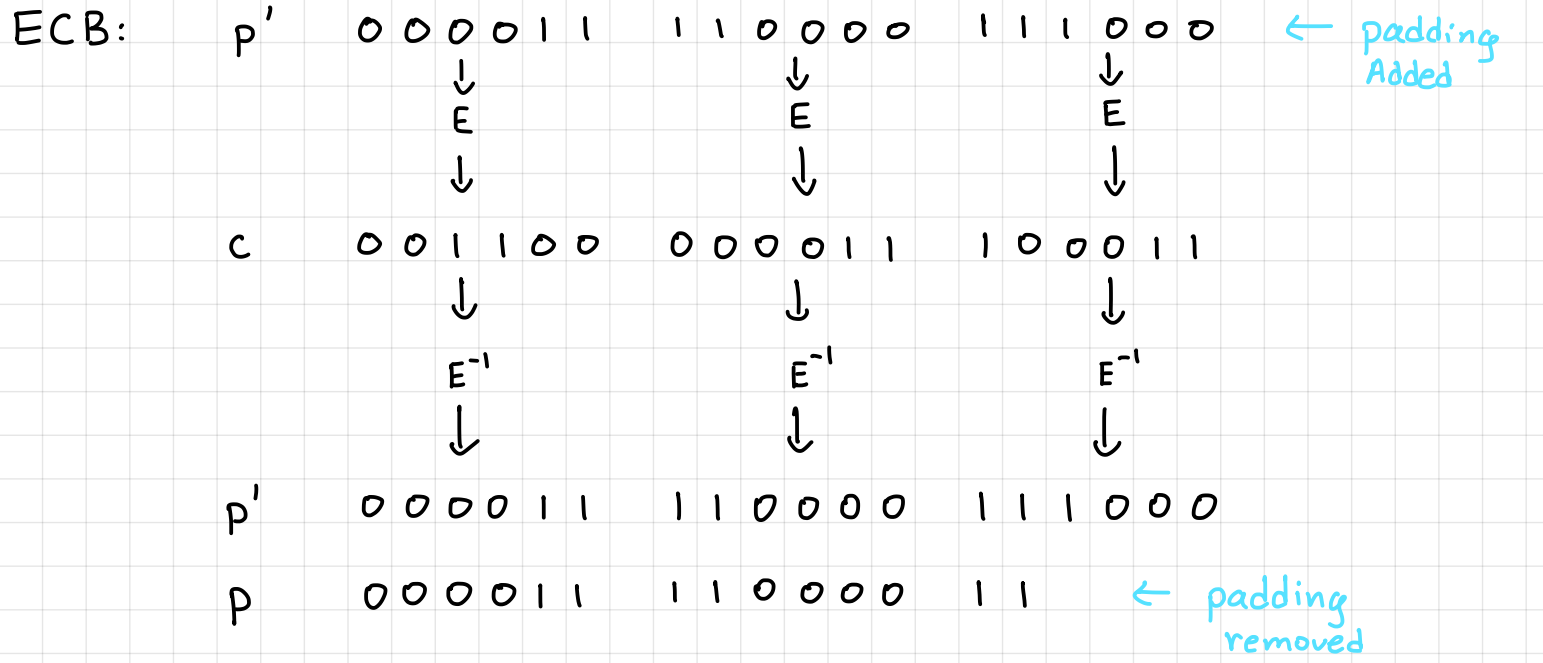
Note:

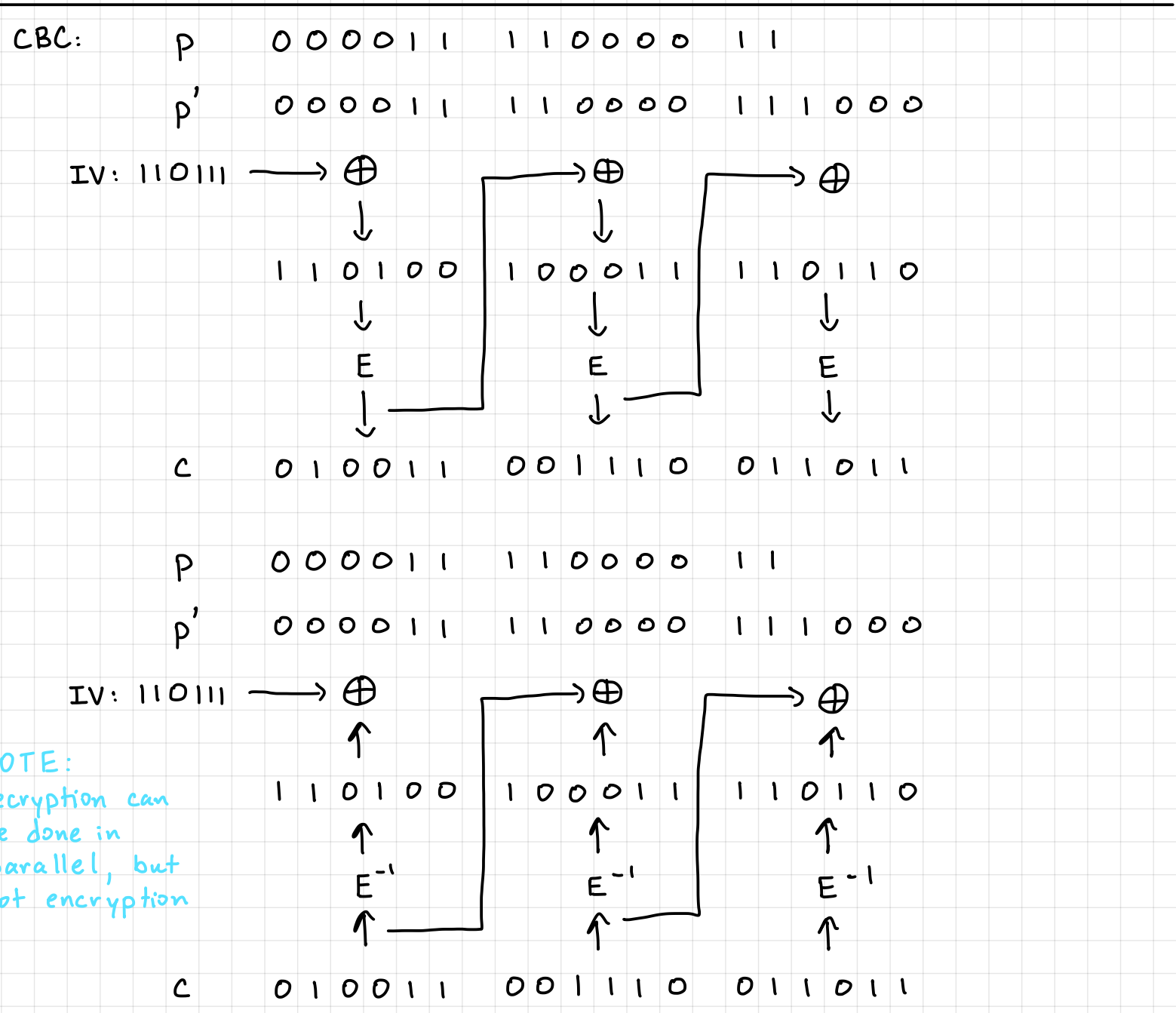
ECB: Can Encrypt and Decrypt in parallel

CBC: Decryption can be done in parallel, but Encryption cannot be done in parallel

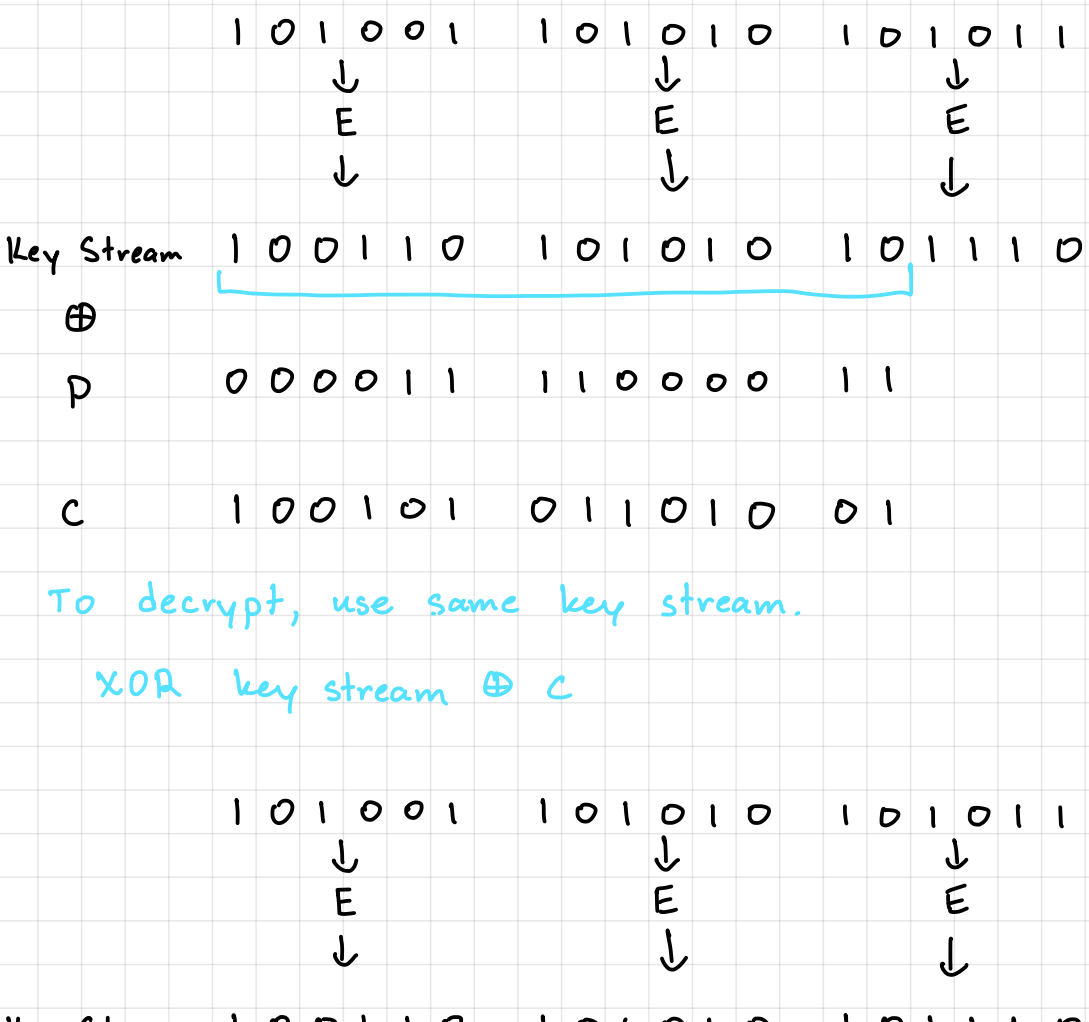
CTR: Can Encrypt and Decrypt in parallel

OFB: Cannot do either Encryption or Decryption in parallel


ECB:
p'
000011 110000 111000
← padding Added
E
c
001100 000011 100011
E⁻¹
p'
000011 110000 111000
p
000011 110000 11
← padding removed

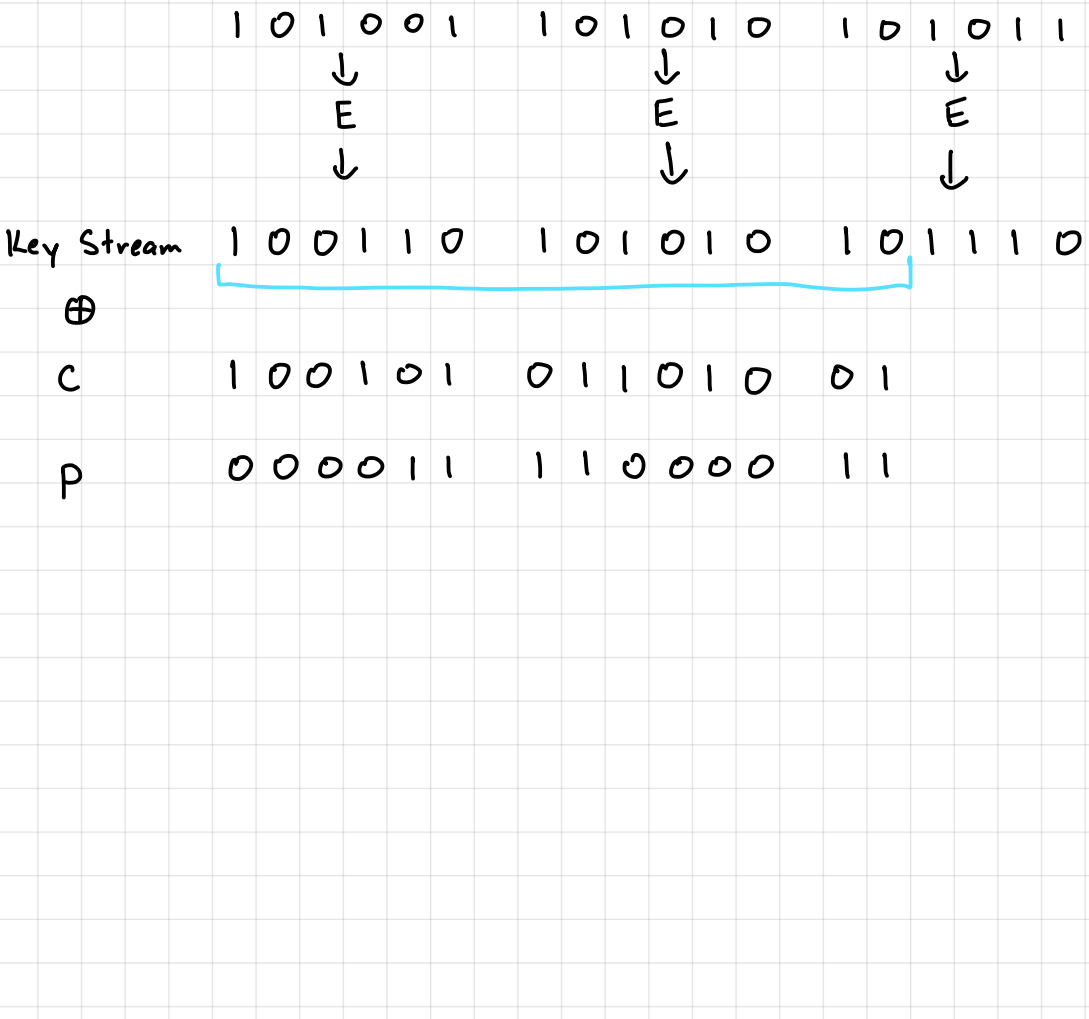

CBC:
p
000011 110000 11
p'
000011 110000 111000
IV: 110111
110100 100011 110110
E
c
010011 001110 011011
p
000011 110000 11
p'
000011 110000 111000
IV: 110111
110100 100011 110110
E⁻¹
c
010011 001110 011011
NOTE:
Decryption can be done in parallel, but not encryption

CTR:

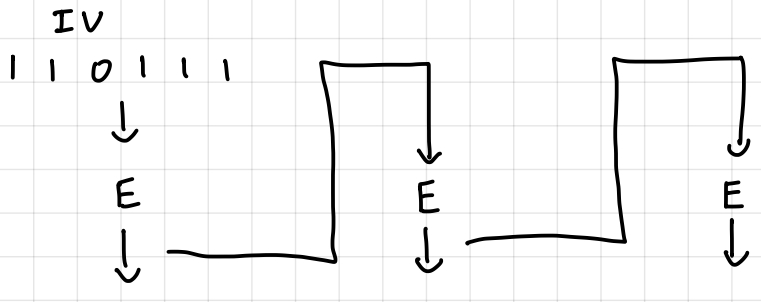


To decrypt, use same key stream.

XOR key stream ⊕ c

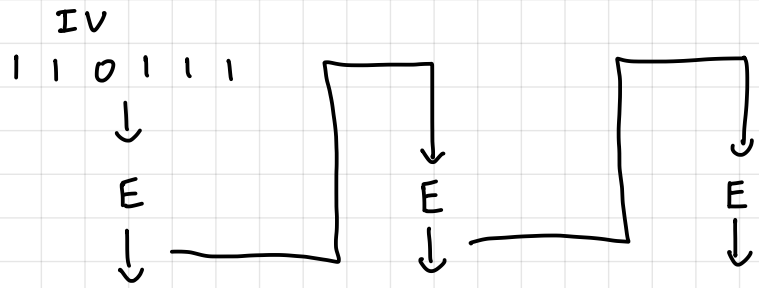


OFB:



Key Stream	011111	111101	110111
⊕			
p	000011	110000	11111
c	011100	001101	0010

TO Decrypt:



Key Stream	011111	111101	110111
⊕			
c	011100	001101	0010
p	000011	110000	11111