Apple II Colour Graphics

Kris Kennaway

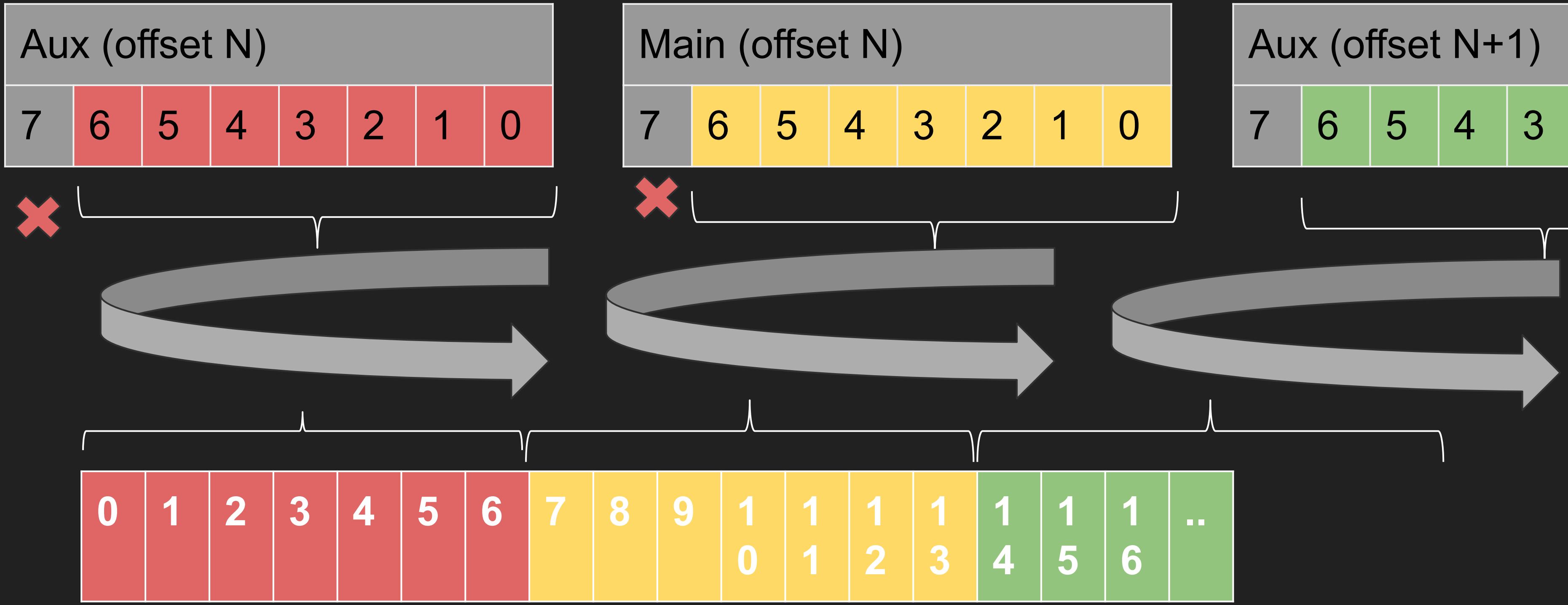
Apple II Colour Graphics

- How colours work in Hi-Res and Double Hi-Res
  - in 5 minutes
- Why all of those crazy rules?
  - e.g. Violet pixel then Green pixel → turns White?!
- ...and even when you follow them, don't get what you expect on the screen
  - fringing, interference

Dots

- Monochrome display
- Start with Double Hi-Res
  - It’s simpler!
    - (said no-one else, ever)
- 560 horizontal dots per line
- High bit of screen byte is ignored
- 7 bits in memory map to 7 dots on screen
- Alternating bytes from AUX, MAIN memory
  - 40 + 40 bytes gives 560 dot line

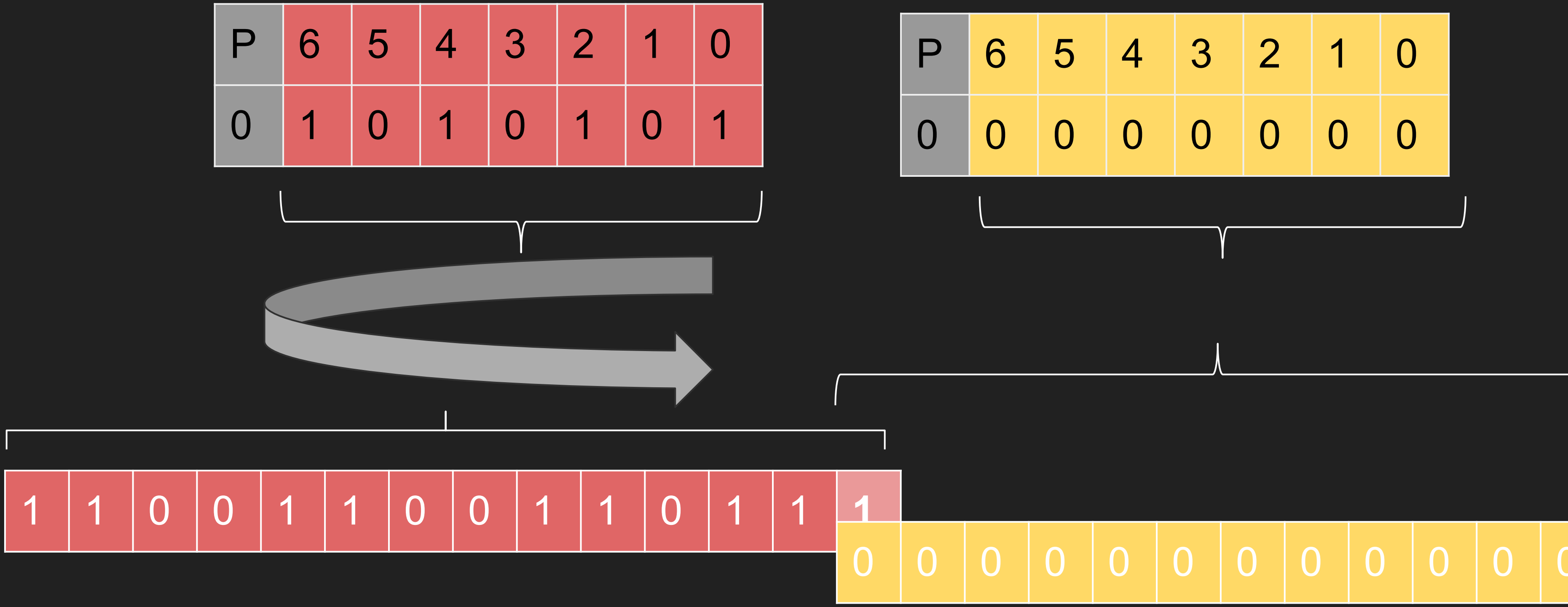
Double Hi-Res dots



Hi-Res dots

- Hi-Res has 280 horizontal resolution, right?
- Nope, also 560
  - but can’t control every dot independently
- Bits 0..5 turn on 2 dots
- Bit 6 turns on **3 dots**
- Third dot may be overwritten by next byte
- Bit 7 (palette bit) shifts dots right by 1 position

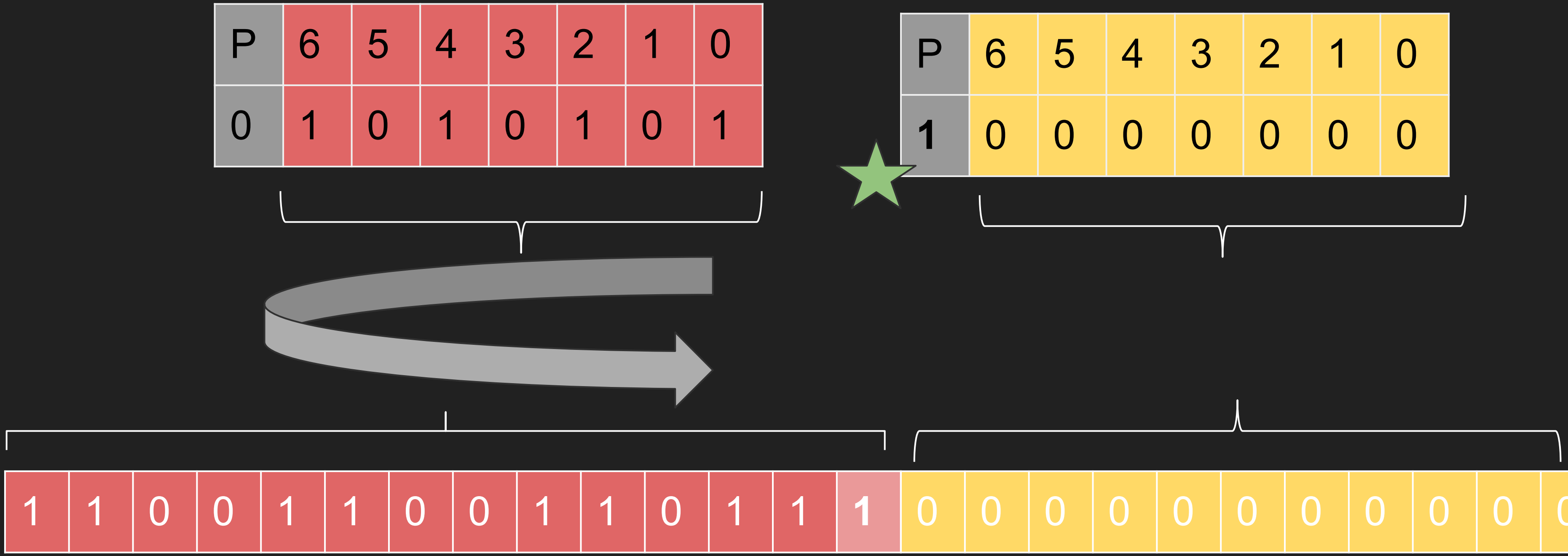
Hi-Res dots



P	6	5	4	3	2	1	0
0	1	0	1	0	1	0	1

P	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Hi-Res dots with palette shift



P	6	5	4	3	2	1	0
0	1	0	1	0	1	0	1

P	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

1	1	0	0	1	1	0	0	1	1	0	1	1	**1**	0	0	0	0	0	0	0	0	0	0

Artifact colours

- Think like a TV
- Scan each line, left to right
- Colour reference signal, 1 complete cycle in same time as displaying 4 dots
- Relative phase of dots determines colour
- Messy and analogue, but simple digital approximation
- **Colour signal sees a sliding 4-bit window of dots**

4-bit colour dot patterns

Hires Colours				
0	0	0	0	
0	0	0	1	🟫
0	0	1	0	🟩
0	0	1	1	🟩
0	1	0	0	🟦
0	1	0	1	⬜
0	1	1	0	🟦
0	1	1	1	🟩

1	0	0	0	🟪
1	0	0	1	🟧
1	0	1	0	⬜
1	0	1	1	🟨
1	1	0	0	🟪
1	1	0	1	🟪
1	1	1	0	🟦
1	1	1	1	⬜

Double Hires Colours				
0	0	0	0	
0	0	1	0	🟫
0	1	0	0	🟩
0	1	1	0	🟩
1	0	0	0	🟦
1	0	1	0	⬜
1	1	0	0	🟦
1	1	1	0	🟩

(left-shifted)				
0	0	0	1	🟪
0	0	1	1	🟧
0	1	0	1	⬜
0	1	1	1	🟨
1	0	0	1	🟪
1	0	1	1	🟪
1	1	0	1	🟦
1	1	1	1	⬜

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3
Bits	**0**	**0**	**0**	**0**
Shifted	**0**	**0**	**0**	**0**
Colour	■			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0
Bits		0	0	0	1
Shifted	**1**	**0**	**0**	**0**	
Colour	⬛	🟪			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0	1
Bits			**0**	**0**	**1**	**1**
Shifted	**1**	**1**	**0**	**0**		
	⬛	🟪	🟪			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0	1	**2**
Bits				**0**	**1**	**1**	**0**
Shifted	**1**	**1**	**0**	**0**			
	⬛	🟪	🟪	🟪			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0	1	**2**	**3**
Bits					**1**	**1**	**0**	**0**
Shifted	**1**	**1**	**0**	**0**				
	⬛	🟪	🟪	🟪	🟪			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0	1	**2**	**3**	**0**
Bits						**1**	**0**	**0**	**0**
Shifted	**0**	**1**	**0**	**0**					
	⬛	🟪	🟪	🟪	🟪	🟦			

Example

Clock	0	1	2	3	0	1	2	3	0	1
Bits	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Clock	0	1	2	3	0	1	**2**	**3**	**0**	**1**
Bits							**0**	**0**	**0**	**0**
Shifted	**0**	**0**	**0**	**0**						
	⬛	🟪	🟪	🟪	🟪	🟦	⬛			

Oh look, we've discovered a Hi-Res Violet pixel (with fringing)

Fringing

Violet

Clock	0	1	2	3	0	1	2	3	0	1	2	3
Bit	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
Colour	⬛	🟪	🟪	🟪	🟪	🟦	⬛	⬛	⬛	?	?	?

+

Green

Clock	0	1	2	3	0	1	2	3	0	1	2	3
Bit	0	0	0	0	0	0	1	1	0	0	0	0
Colour	⬛	⬛	⬛	🟩	🟩	🟩	🟩	🟫		?	?	?

=

Interference

White

Clock	0	1	2	3	0	1	2	3	0	1	2	3
Bit	0	0	0	0	1	1	1	1	0	0	0	0
Colour	⬛	🟪	🟪	🟦	⬜	🟩	🟩	🟫		?	?	?

Everyone knows there are only 6 Hi-Res Colours

- unless you read Sather, “Understanding the Apple IIe”
- Remember the funny business with the palette bit shifting dots by 1 position, and how this extends/truncates dot patterns at the byte boundary?
- Can get 14 of 16 colours at byte boundaries
    - Plus the usual fringing

Hi-Res Yellow?!

Odd offset							
0	1	2	3	4	5	6	P
0	0	0	0	0	0	1	0

Even offset							
0	1	2	3	4	5	6	P
0	0	0	0	0	0	0	**1**

Bit Pos	4	4	5	5	6	6	**6**	0	0	1	1
Clock	2	3	0	1	2	3	0	1	2	3	0
Bit	0	0	0	0	1	1	1	0	0	0	0
Colour	⬛	🟩	🟩	🟨	🟨	🟧	🟪	⬛	?	?	?