Inside A2Stream

glitch-free
high-quality
internet audio
streaming
on the Apple II

Oliver Schmidt
KanasFest 2022

Demo

Requirements:

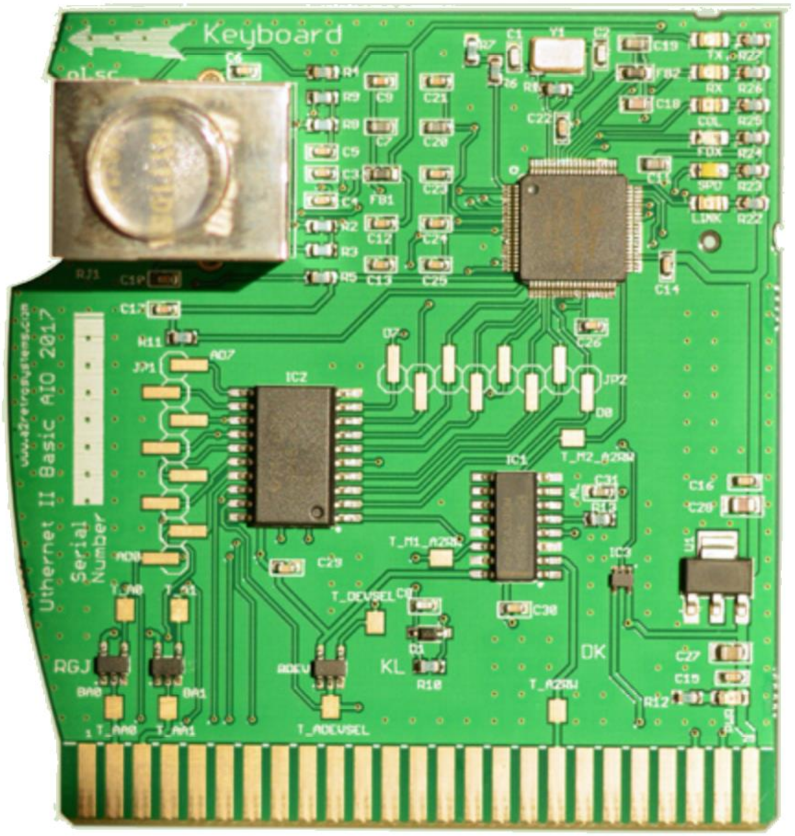
- Enhanced //e incl. ext. 80 column card

  or

- IIgs

  and

- a2RetroSystems Uthernet II card



Background

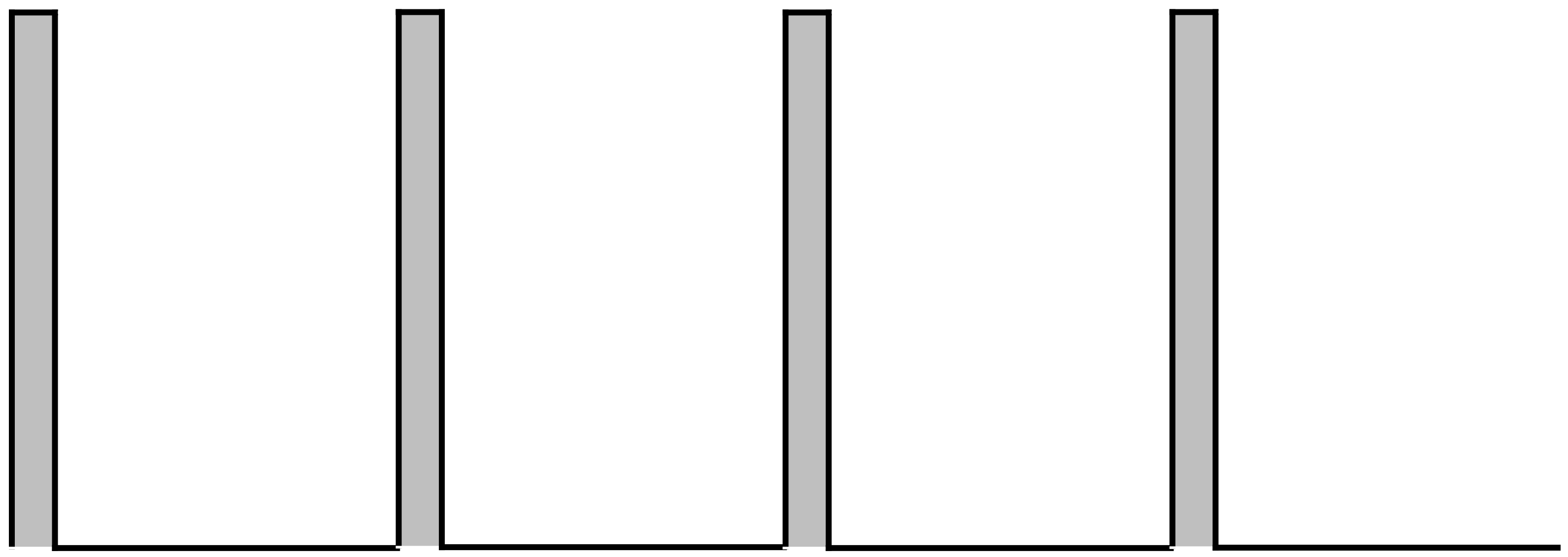
- A2Stream is built around pulse-width modulation (PWM)
- Why not pulse-density modulation (PDM)?

- Complete, working implementation based on PDM
- Quality not satisfying

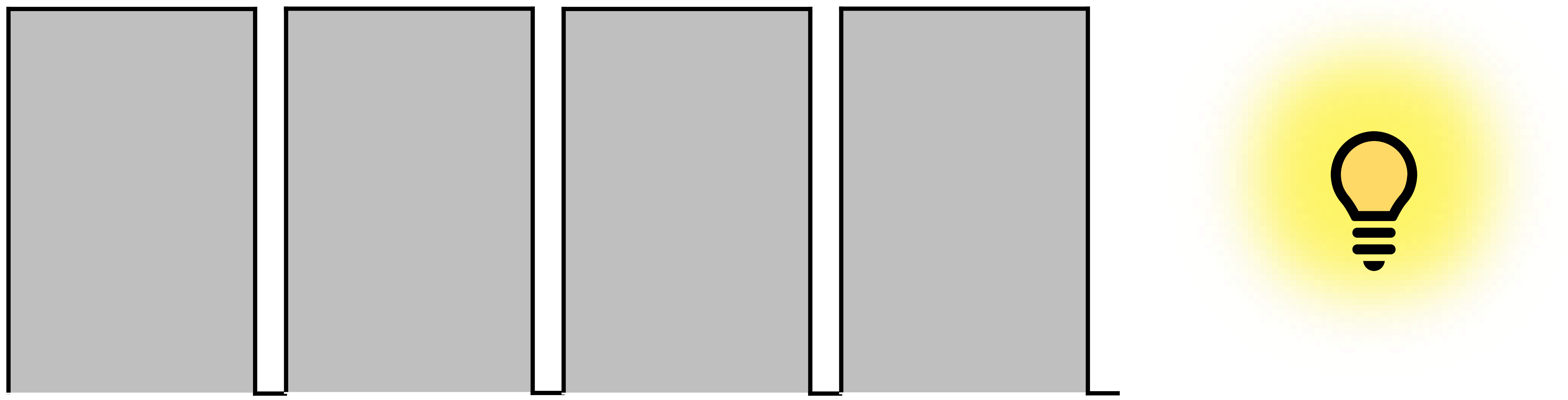
- Michael J. Mahon explained why he considers PWM superior

So here we are…

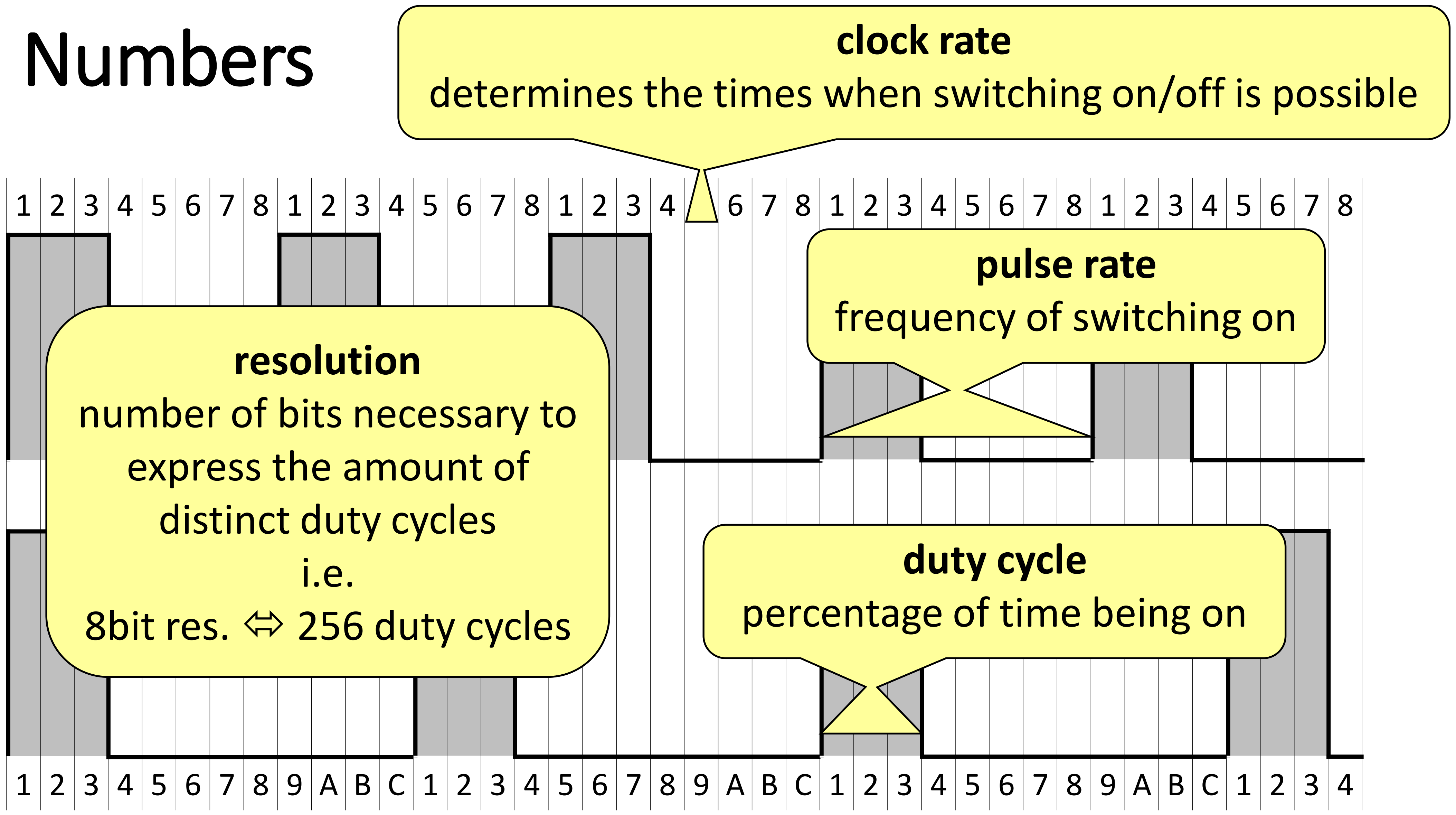
Pulse-Width Modulation



Pulse-Width Modulation



Numbers


1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9 A B C 1 2 3 4 5 6 7 8 9 A B C 1 2 3 4 5 6 7 8 9 A B C 1 2 3 4

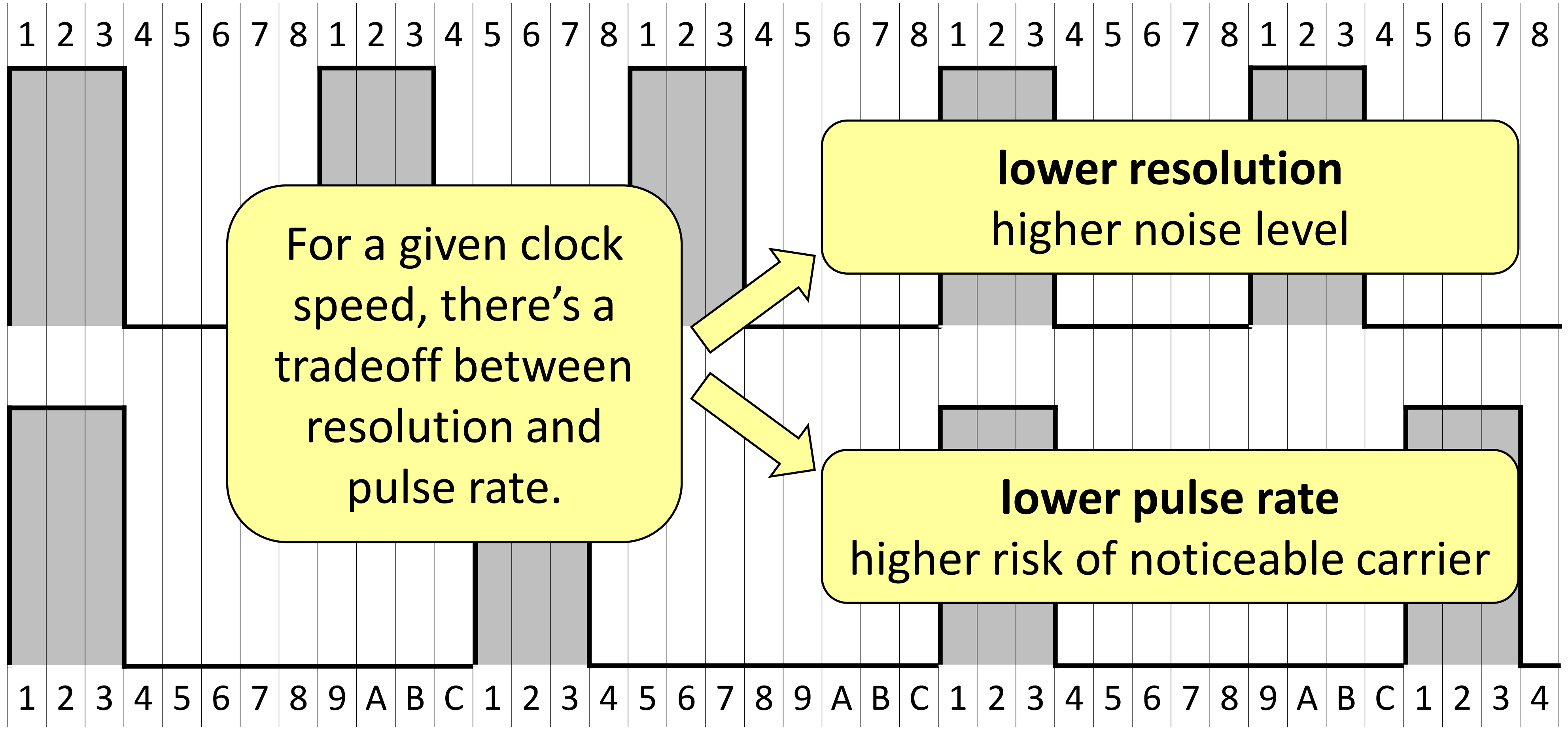
clock rate
determines the times when switching on/off is possible

pulse rate
frequency of switching on

resolution
number of bits necessary to express the amount of distinct duty cycles
i.e.
8bit res. ⇔ 256 duty cycles

duty cycle
percentage of time being on

Tradeoff


1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9 A B C 1 2 3 4 5 6 7 8 9 A B C 1 2 3 4 5 6 7 8 9 A B C 1 2 3 4

For a given clock speed, there's a tradeoff between resolution and pulse rate.

lower resolution
higher noise level

lower pulse rate
higher risk of noticeable carrier

Values for A2Stream

- Target pulse rate: 22050 Hz
  - Same as DAC522 … no more experiments ;-)

- A2Stream uses individual machine cycles as clock rate
  - Although all 6502 instructions require at least two machine cycles

- So, what is the possible resolution for the target pulse rate?
  - With 1 MHz, there are 46 machine cycles per pulse
  - But that does **not** mean 46 distinct duty cycles…

A2Stream Duty Cycles

Shortest duty cycle:

```
; 1. speaker toggle
STA $C030              4
; 2. speaker toggle
STA $C030              4
; spend some cycles
…                     35
JMP somewhere          3
                     ___
                      46
```

Longest duty cycle:

```
; 1. speaker toggle
STA $C030              4
; spend some cycles
…                     35
; 2. speaker toggle
STA $C030              4
JMP somewhere          3
                     ___
                      46
```

A2Stream Duty Cycles

Shortest duty cycle:

The first few duty cycles are impossible

```
                 oggle
                                4
; 2. speaker toggle
STA $C030
; spend some cycles
…                              35
JMP somewhere                   3
                              ___
                               46
```

A2Stream implements **36** distinct duty cycles in the 46 machine cycles

Longest duty cycle:

The last few duty cycles are impossible

```
; 1. speaker t
STA $C030
            some cycles
                                35
            aker toggle
STA $C030                        4
JMP
```

BTW: DAC522 implements **32** distinct duty cycles in the same 46 machine cycles

Pulse Generators

- For the 36 distinct duty cycles, there are 36 distinct pieces of code
- Every execution of one of those pieces generates exactly one pulse

- Running the same pulse generator several times, keeps the speaker
  - Like the PWM-controlled light bulb keeping the brightness
- Running different pulse generators in a sequence, moves the speaker
  - Like the PWM-controlled light bulb fading in/out

- The task of A2Stream is to run the 36 pulse generators in a sequence that reflects the changes in the audio signal

JMP Target Modification

- The 6502 is 8-bit, but its address space is 16-bit
- The 6502 address space can be seen as 256 pages of 256 bytes each
- Each address consists of 2 bytes
  - The high-byte denotes the page, the low-byte denotes the offset on that page
- The JMP instruction consists of 3 bytes:

```
JMP OpCode,Target Offset,Target Page
```

- A2Stream modifies every JMP target before executing it
  - Self-modifying code

Pulse Generator Placement

- A2Stream can’t spend the cycles to modify both bytes of a JMP target
- Option 1: All pulse generators are on the same page ⇨ doesn’t fit ✘
- Option 2: All pulse generators are on the same offset ✓

⇨ The 36 pulse generators are on the 36 pages \$40 – \$63

- Pages \$20 – \$3F are used by the high resolution graphics screen

⇨ „All“ a pulse generator has to do with the cycles just spent so far:

1. Get somehow a byte with a value \$40 – \$63
2. Set the JMP target high byte to that value

Uthernet II Network Interface

- The *a2RetroSystems Uthernet II* card is build around the WIZnet W5100 Ethernet controller
- The W5100 contains a TCP/IP stack, but most Apple II programs rather use their own TCP/IP stack
- Usually, a web client wants to receive the data (much) faster than a web server can send it
- In contrast, A2Stream needs to receive the data (much) slower than a web server wants to send it ⇨ the web server needs to be throttled
- A2Stream fully relies on the W5100 TCP/IP stack to autonomously take care of TCP flow control, which provides that very throttling

Uthernet II Host Interface

- Receiving TCP data with the W5100 means to:
    1. Check, if there’s data available in the W5100
    2. Compute the W5100 RAM address of the data
    3. Read the individual data bytes from the W5100
    4. Commit the number of bytes read to the W5100
- Steps 1.), 2.) and 4.) all require way more cycles than the ones left in a single pulse generator ⇨ there’s no data during those steps :-(

Solving this problem is *THE* innovation of A2Stream!

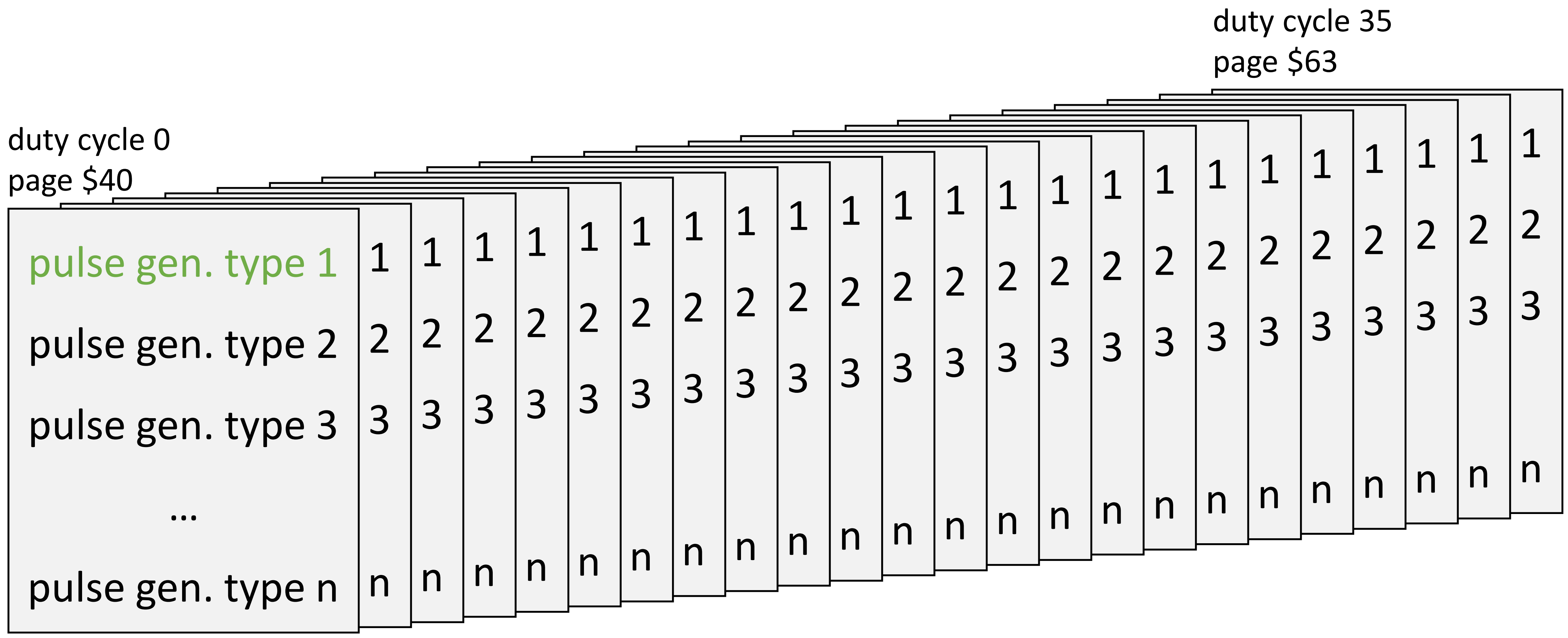
Pulse Generator Types

- The individual steps are distributed over different pulse generators
- There’s a 2-dimensional array of pulse generators:
    1. dimension: The 36 distinct duty cycles
    2. dimension: The different pulse generator types

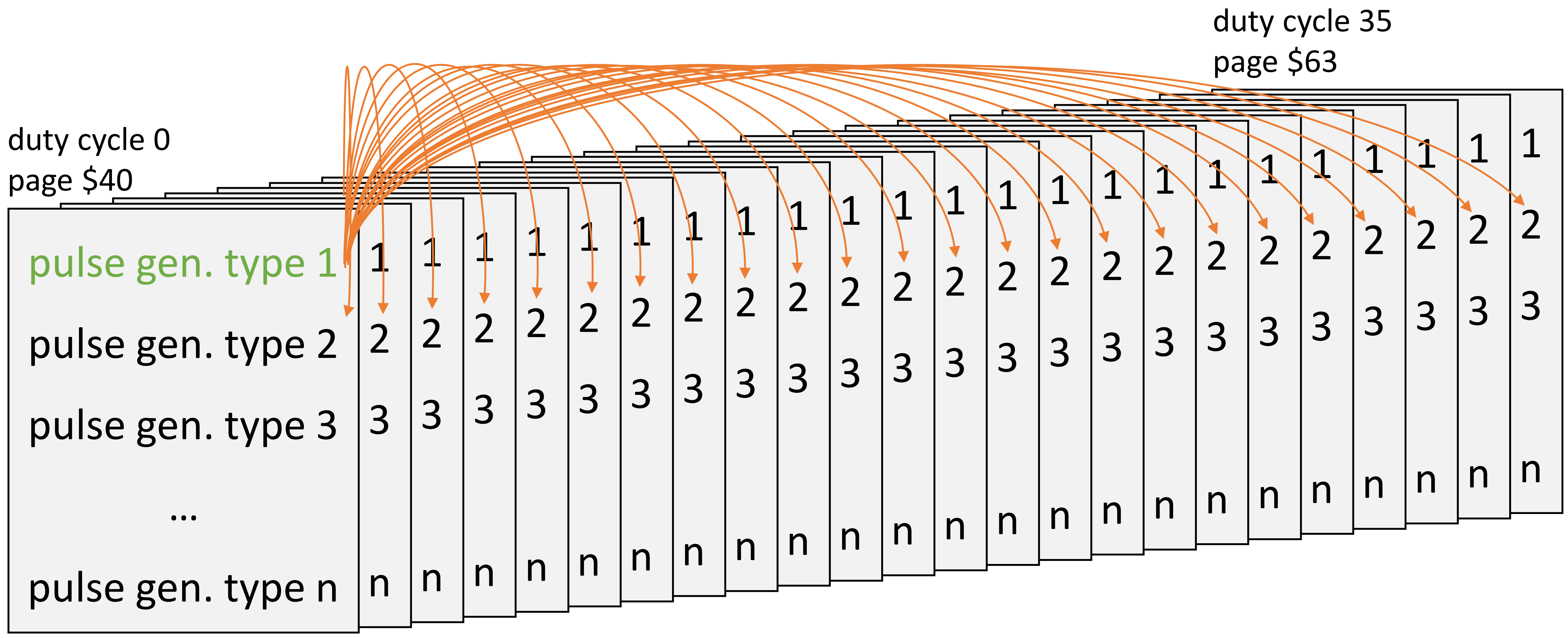
- Fortunately, pulse generators are short enough to share pages

⇨ The (unmodified) JMP target low byte can control the sequence of the pulse generator types – and this way the sequence of the necessary steps to take

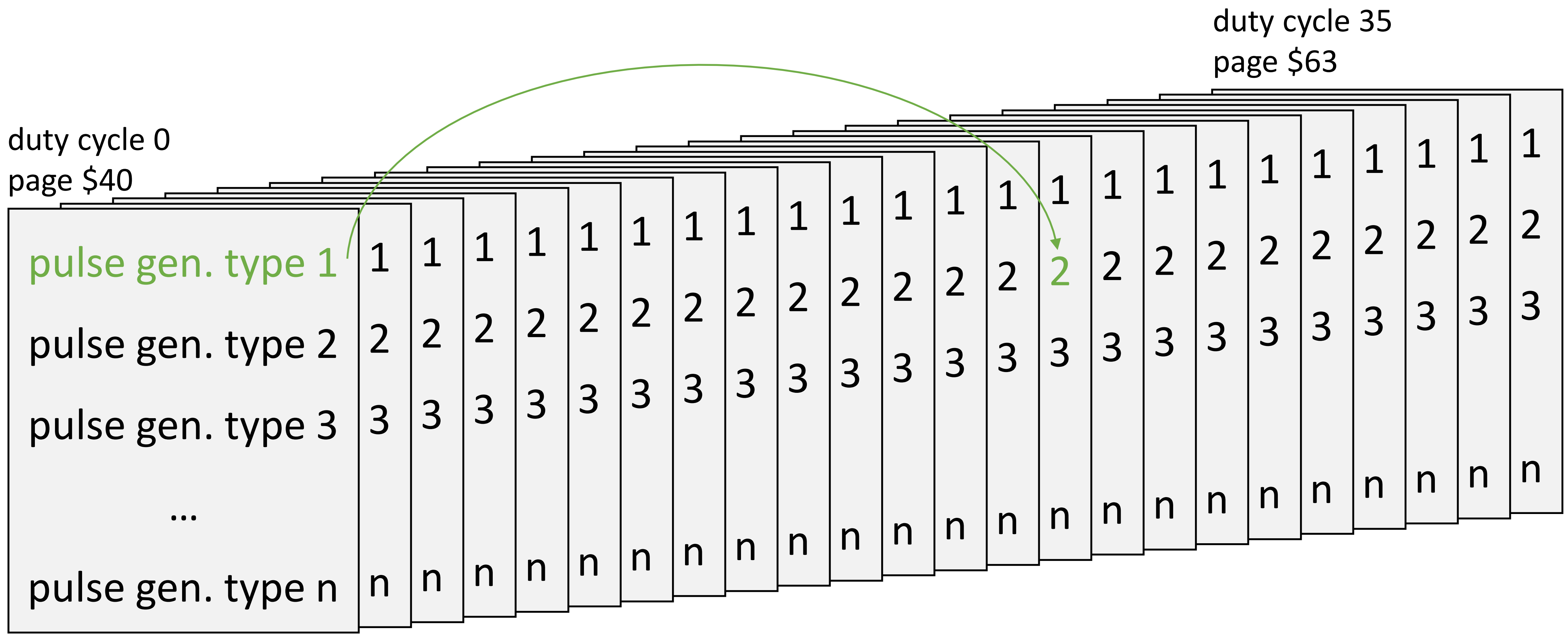
Pulse Generator Program Flow



Pulse Generator Program Flow



Pulse Generator Program Flow



Ring Buffer

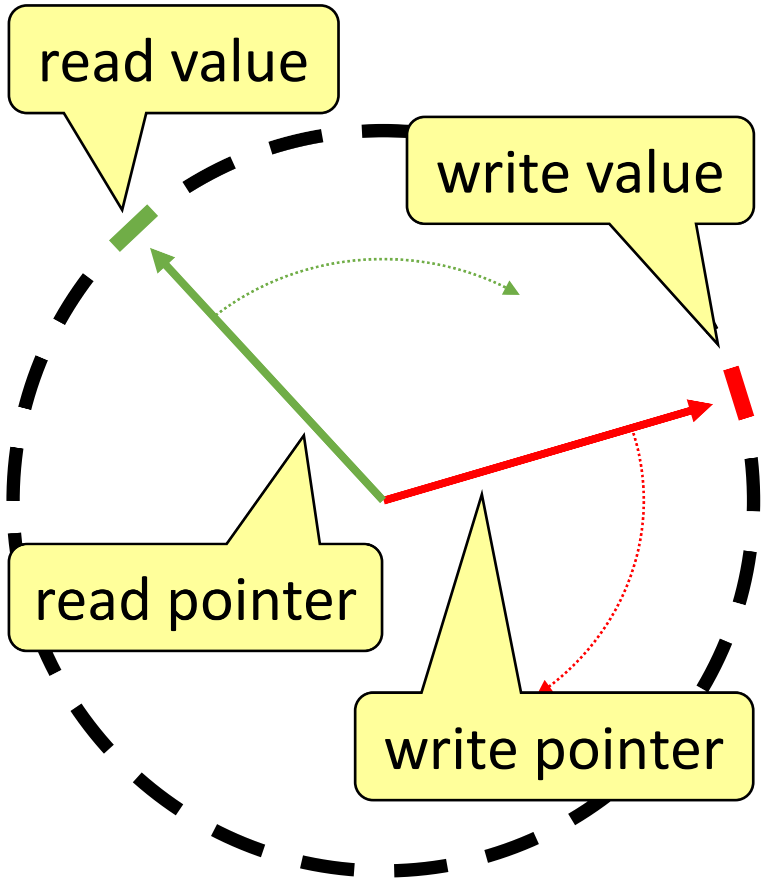
- The W5100 can’t provide a contiguous byte stream

⇨ A2Strem needs a ring buffer

- Every pulse generator reads exactly one byte from the buffer ⇨ JMP
- Some pulse generator types can’t write a byte into the buffer ⇨ W5100

⇨ At least one pulse generator type must write more than one byte into the buffer

Stack Pointer



- 6502 has only 3 general registers (A, X, Y)
- Abuse the stack pointer (SP) as 4. register
- 65C02-specific instructions required

	SP: read pointer	SP: write pointer

write	`STA $0100,Y`
	`INY`
		`PHX` (PUSH X to stack)

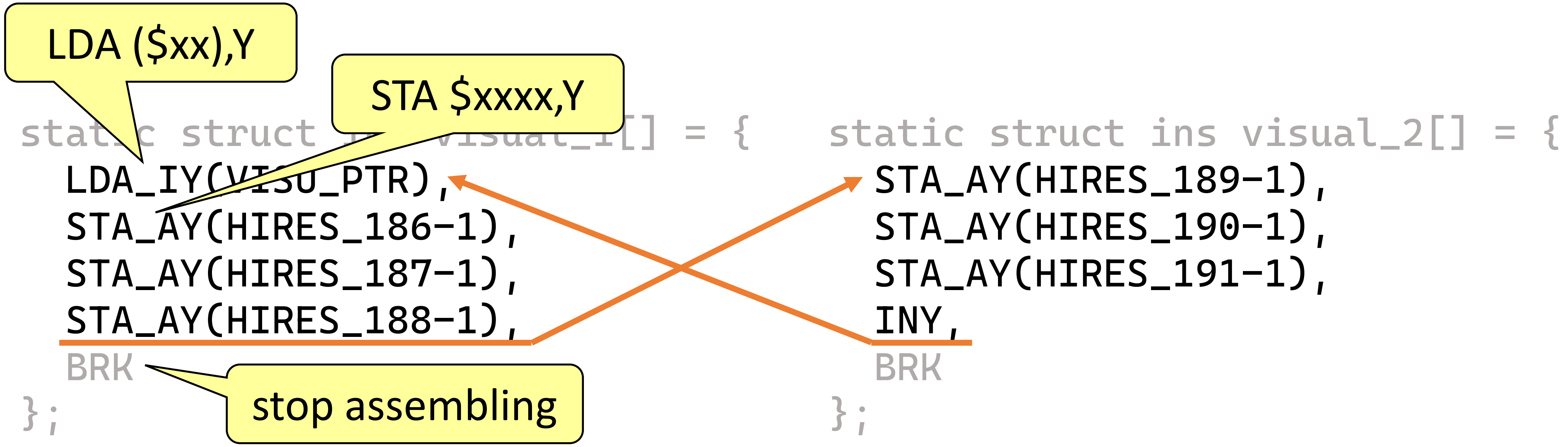
read	`PLX` (PULL X from stack)	`LDA $0100,Y`
		`DEY`

Custom Assembler

- Even with abusing SP, a single pulse generator still can’t write 2 bytes into the ring buffer :-(
- However, 2 pulse generators together – of course of different types – manage to write 3 bytes :-)
- A2Stream creates the code for all those pulse generators at runtime
  - Allow to select Uthernet II slot
  - Allow to select speaker ($C030) vs. tape out ($C020)
- A2Stream contains a custom assembler written in C using cc65
  - Only one source for all duty cycles (instruction scheduling is automatic)
  - Only the source that differs between pulse generator types (other is implicit)

Source Code

- Source of two pulse generator types used for the visualization
- Those two types are executed by turns
- The code can presume both A and Y to be retained!



```
static struct ins visual_1[] = {
   LDA_IY(VISU_PTR),
   STA_AY(HIRES_186-1),
   STA_AY(HIRES_187-1),
   STA_AY(HIRES_188-1),
   BRK
};
```

```
static struct ins visual_2[] = {
   STA_AY(HIRES_189-1),
   STA_AY(HIRES_190-1),
   STA_AY(HIRES_191-1),
   INY,
   BRK
};
```

Generated Code

cycles
sum is always 46

green
toggle speaker

orange
type-specific

grey
spend cycles

blue
flow control

```
      STA SPEAKER        4
      STA SPEAKER        4
      PLX                4
      STX MOD+2          4
      LDA (ZP_VISUAL),Y  5
      STA HIRES_186-1,Y  5
      STA HIRES_187-1,Y  5
      STA HIRES_188-1,Y  5
      BRA NXT            3
NXT:  NOP                2
      NOP                2
MOD:  JMP OFFSET         3
```

Duty Cycles 00 and 01

65C02-specific instruction like STA ($xx),Y but without Y

```
      STA SPEAKER           4
      STA SPEAKER           4
      PLX                   4
      STX MOD+2             4
      LDA (ZP_VISUAL),Y     5
      STA HIRES_186-1,Y     5
      STA HIRES_187-1,Y     5
      STA HIRES_188-1,Y     5
      BRA NXT               3
NXT:  NOP                   2
      NOP                   2
MOD:  JMP OFFSET            3
```

```
      STA SPEAKER           4
      STA (ZP_SPEAKER)      5
      PLX                   4
      STX MOD+2             4
      LDA (ZP_VISUAL),Y     5
      STA HIRES_186-1,Y     5
      STA HIRES_187-1,Y     5
      STA HIRES_188-1,Y     5
      NOP                   2
      NOP                   2
      NOP                   2
MOD:  JMP OFFSET            3
```

Duty Cycles 02 and 03

```
      STA SPEAKER         4
      NOP                 2
      STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      BRA NXT             3
NXT:  NOP                 2
MOD:  JMP OFFSET          3
```

```
      STA SPEAKER         4
      BRA NXT             3
NXT:  STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      NOP                 2
      NOP                 2
MOD:  JMP OFFSET          3
```

Duty Cycles 04 and 05

```
      STA SPEAKER         4
      PLX                 4
      STA SPEAKER         4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      BRA NXT             3
NXT:  NOP                 2
      NOP                 2
MOD:  JMP OFFSET          3
```

```
      STA SPEAKER         4
      PLX                 4
      STA (ZP_SPEAKER)    5
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      NOP                 2
      NOP                 2
      NOP                 2
MOD:  JMP OFFSET          3
```

Duty Cycles 06 and 07

```
     STA SPEAKER        4
     PLX                4
     NOP                2
     STA SPEAKER        4
     STX MOD+2          4
     LDA (ZP_VISUAL),Y  5
     STA HIRES_186-1,Y  5
     STA HIRES_187-1,Y  5
     STA HIRES_188-1,Y  5
     BRA NXT            3
NXT: NOP                2
MOD: JMP OFFSET         3
```

```
     STA SPEAKER        4
     PLX                4
     BRA NXT            3
NXT: STA SPEAKER        4
     STX MOD+2          4
     LDA (ZP_VISUAL),Y  5
     STA HIRES_186-1,Y  5
     STA HIRES_187-1,Y  5
     STA HIRES_188-1,Y  5
     NOP                2
     NOP                2
MOD: JMP OFFSET         3
```

Duty Cycles 08 and 09

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      STA SPEAKER          4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA NXT              3
NXT:  NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      STA (ZP_SPEAKER)     5
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      NOP                  2
      NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

Duty Cycles 10 and 11

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      NOP                  2
      STA SPEAKER          4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA NXT              3
NXT:  NOP                  2
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      BRA NXT              3
NXT:  STA SPEAKER          4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

Duty Cycles 12 and 13

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      NOP                  2
      NOP                  2
      STA SPEAKER          4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA MOD              3
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA SPEAKER          4
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA NXT              3
NXT:  NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

Duty Cycles 14 and 15

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA (ZP_SPEAKER)     5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      NOP                  2
      NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      NOP                  2
      STA SPEAKER          4
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA NXT              3
NXT:  NOP                  2
MOD:  JMP OFFSET           3
```

Duty Cycles 16 and 17

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      BRA NXT              3
NXT:  STA SPEAKER          4
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      NOP                  2
      NOP                  2
      STA SPEAKER          4
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA MOD              3
MOD:  JMP OFFSET           3
```

Duty Cycles 18 and 19

```
     STA SPEAKER          4
     PLX                  4
     STX MOD+2            4
     LDA (ZP_VISUAL),Y    5
     STA HIRES_186-1,Y    5
     STA SPEAKER          4
     STA HIRES_187-1,Y    5
     STA HIRES_188-1,Y    5
     BRA NXT              3
NXT: NOP                  2
     NOP                  2
MOD: JMP OFFSET           3
```

```
     STA SPEAKER          4
     PLX                  4
     STX MOD+2            4
     LDA (ZP_VISUAL),Y    5
     STA HIRES_186-1,Y    5
     STA (ZP_SPEAKER)     5
     STA HIRES_187-1,Y    5
     STA HIRES_188-1,Y    5
     NOP                  2
     NOP                  2
     NOP                  2
MOD: JMP OFFSET           3
```

Duty Cycles 20 and 21

```
     STA SPEAKER          4
     PLX                  4
     STX MOD+2            4
     LDA (ZP_VISUAL),Y    5
     STA HIRES_186-1,Y    5
     NOP                  2
     STA SPEAKER          4
     STA HIRES_187-1,Y    5
     STA HIRES_188-1,Y    5
     BRA NXT              3
NXT: NOP                  2
MOD: JMP OFFSET           3
```

```
     STA SPEAKER          4
     PLX                  4
     STX MOD+2            4
     LDA (ZP_VISUAL),Y    5
     STA HIRES_186-1,Y    5
     BRA NXT              3
NXT: STA SPEAKER          4
     STA HIRES_187-1,Y    5
     STA HIRES_188-1,Y    5
     NOP                  2
     NOP                  2
MOD: JMP OFFSET           3
```

Duty Cycles 22 and 23

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      NOP                  2
      NOP                  2
      STA SPEAKER          4
      STA HIRES_187-1,Y    5
      STA HIRES_188-1,Y    5
      BRA MOD              3
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      STA SPEAKER          4
      STA HIRES_188-1,Y    5
      BRA NXT              3
NXT:  NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

Duty Cycles 24 and 25

```
        STA SPEAKER          4
        PLX                  4
        STX MOD+2            4
        LDA (ZP_VISUAL),Y    5
        STA HIRES_186-1,Y    5
        STA HIRES_187-1,Y    5
        STA (ZP_SPEAKER)     5
        STA HIRES_188-1,Y    5
        NOP                  2
        NOP                  2
        NOP                  2
MOD:    JMP OFFSET           3
```

```
        STA SPEAKER          4
        PLX                  4
        STX MOD+2            4
        LDA (ZP_VISUAL),Y    5
        STA HIRES_186-1,Y    5
        STA HIRES_187-1,Y    5
        NOP                  2
        STA SPEAKER          4
        STA HIRES_188-1,Y    5
        BRA NXT              3
NXT:    NOP                  2
MOD:    JMP OFFSET           3
```

Duty Cycles 26 and 27

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      BRA NXT              3
NXT:  STA SPEAKER          4
      STA HIRES_188-1,Y    5
      NOP                  2
      NOP                  2
MOD:  JMP OFFSET           3
```

```
      STA SPEAKER          4
      PLX                  4
      STX MOD+2            4
      LDA (ZP_VISUAL),Y    5
      STA HIRES_186-1,Y    5
      STA HIRES_187-1,Y    5
      NOP                  2
      NOP                  2
      STA SPEAKER          4
      STA HIRES_188-1,Y    5
      BRA MOD              3
MOD:  JMP OFFSET           3
```

Duty Cycles 28 and 29

```
      STA SPEAKER            4
      PLX                    4
      STX MOD+2              4
      LDA (ZP_VISUAL),Y      5
      STA HIRES_186-1,Y      5
      STA HIRES_187-1,Y      5
      STA HIRES_188-1,Y      5
      STA SPEAKER            4
      BRA NXT                3
NXT:  NOP                    2
      NOP                    2
MOD:  JMP OFFSET             3
```

```
      STA SPEAKER            4
      PLX                    4
      STX MOD+2              4
      LDA (ZP_VISUAL),Y      5
      STA HIRES_186-1,Y      5
      STA HIRES_187-1,Y      5
      STA HIRES_188-1,Y      5
      STA (ZP_SPEAKER)       5
      NOP                    2
      NOP                    2
      NOP                    2
MOD:  JMP OFFSET             3
```

Duty Cycles 30 and 31

```
      STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      NOP                 2
      STA SPEAKER         4
      BRA NXT             3
NXT:  NOP                 2
MOD:  JMP OFFSET          3
```

```
      STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      BRA NXT             3
NXT:  STA SPEAKER         4
      NOP                 2
      NOP                 2
MOD:  JMP OFFSET          3
```

Duty Cycles 32 and 33

```
      STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      NOP                 2
      NOP                 2
      STA SPEAKER         4
      BRA MOD             3
MOD:  JMP OFFSET          3
```

```
      STA SPEAKER         4
      PLX                 4
      STX MOD+2           4
      LDA (ZP_VISUAL),Y   5
      STA HIRES_186-1,Y   5
      STA HIRES_187-1,Y   5
      STA HIRES_188-1,Y   5
      BRA NXT             3
NXT:  NOP                 2
      STA SPEAKER         4
      NOP                 2
MOD:  JMP OFFSET          3
```

Duty Cycles 34 and 35

```
      STA  SPEAKER            4
      PLX                     4
      STX  MOD+2              4
      LDA  (ZP_VISUAL),Y      5
      STA  HIRES_186-1,Y      5
      STA  HIRES_187-1,Y      5
      STA  HIRES_188-1,Y      5
      BRA  MOD                3
MOD:  JMP  OFFSET-5           3
      STA  SPEAKER            4
      NOP                     2
      NOP                     2
```

```
      STA  SPEAKER            4
      NOP                     2
      NOP                     2
      STA  SPEAKER            4
      PLX                     4
      STX  MOD+2              4
      LDA  (ZP_VISUAL),Y      5
      STA  HIRES_186-1,Y      5
      STA  HIRES_187-1,Y      5
      STA  HIRES_188-1,Y      5
      BRA  NXT                3
NXT:  NOP                     2
      NOP                     2
      STA  SPEAKER            4
MOD:  JMP  OFFSET             3
```

Page Sets

- Up to 5 pulse generator types can share a page
- But A2Stream has 10 pulse generators types

⇨ A second set of pages is necessary, it is on the 36 pages $64 – $87

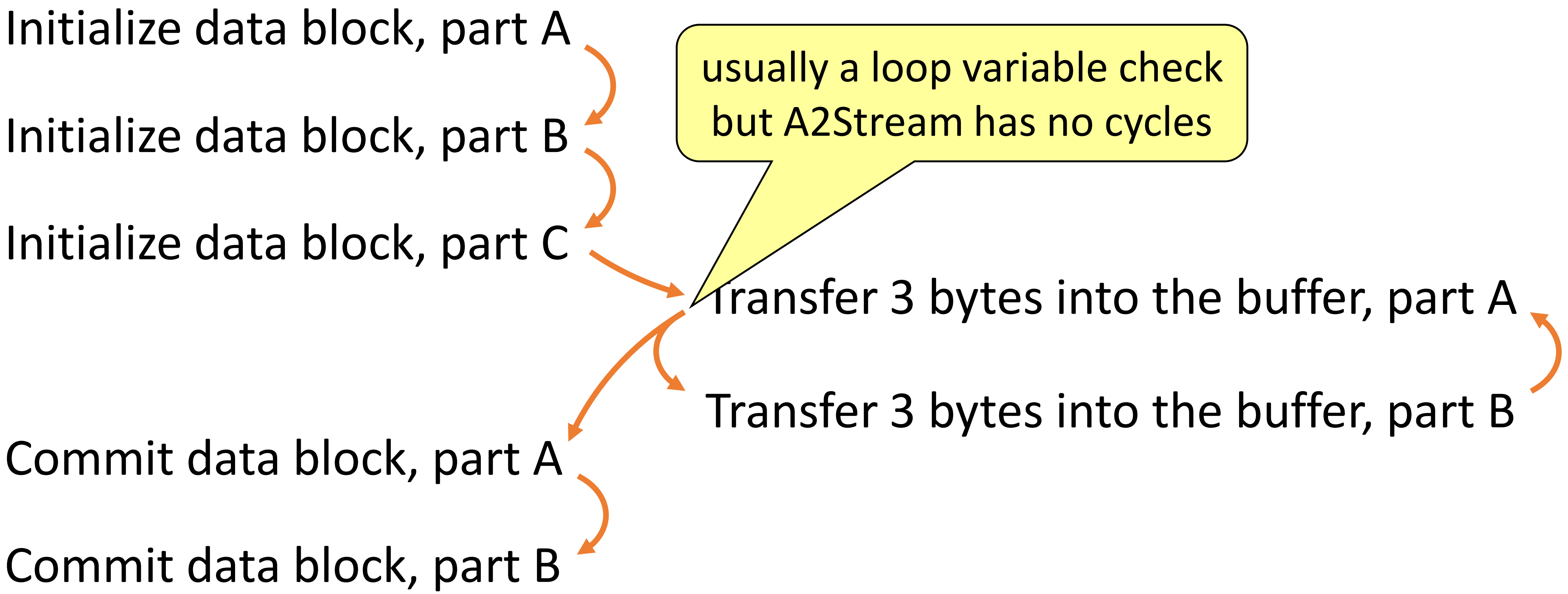
  ⇨ Every pulse generator type is either in page set 0 or in page set 1

⇨ The data needs to contain either $40 – $63 or $64 – $87, depending on the page set, the next pulse generator type is in

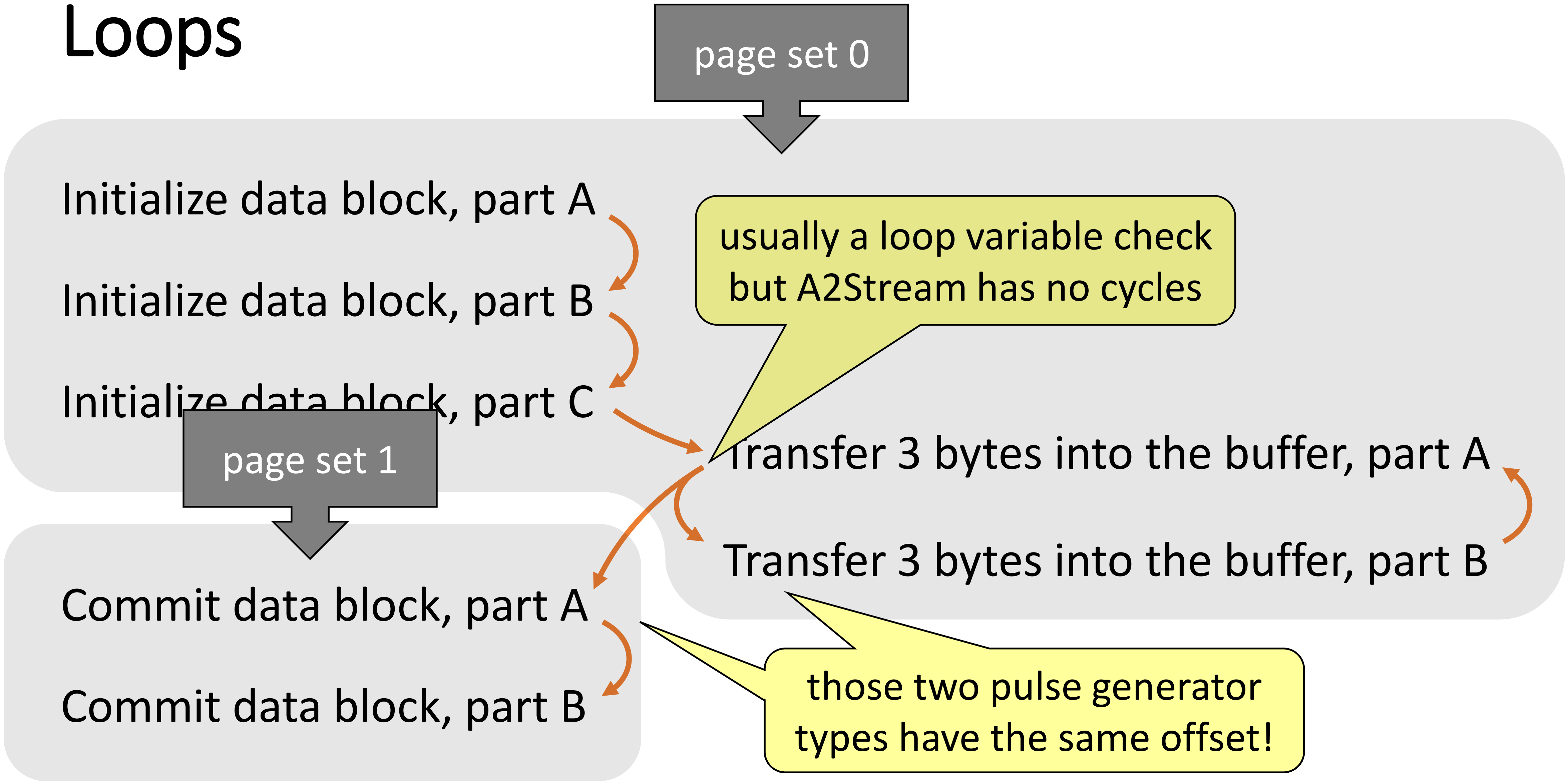
⇨ The data needs to exactly match the internal A2Stream structure

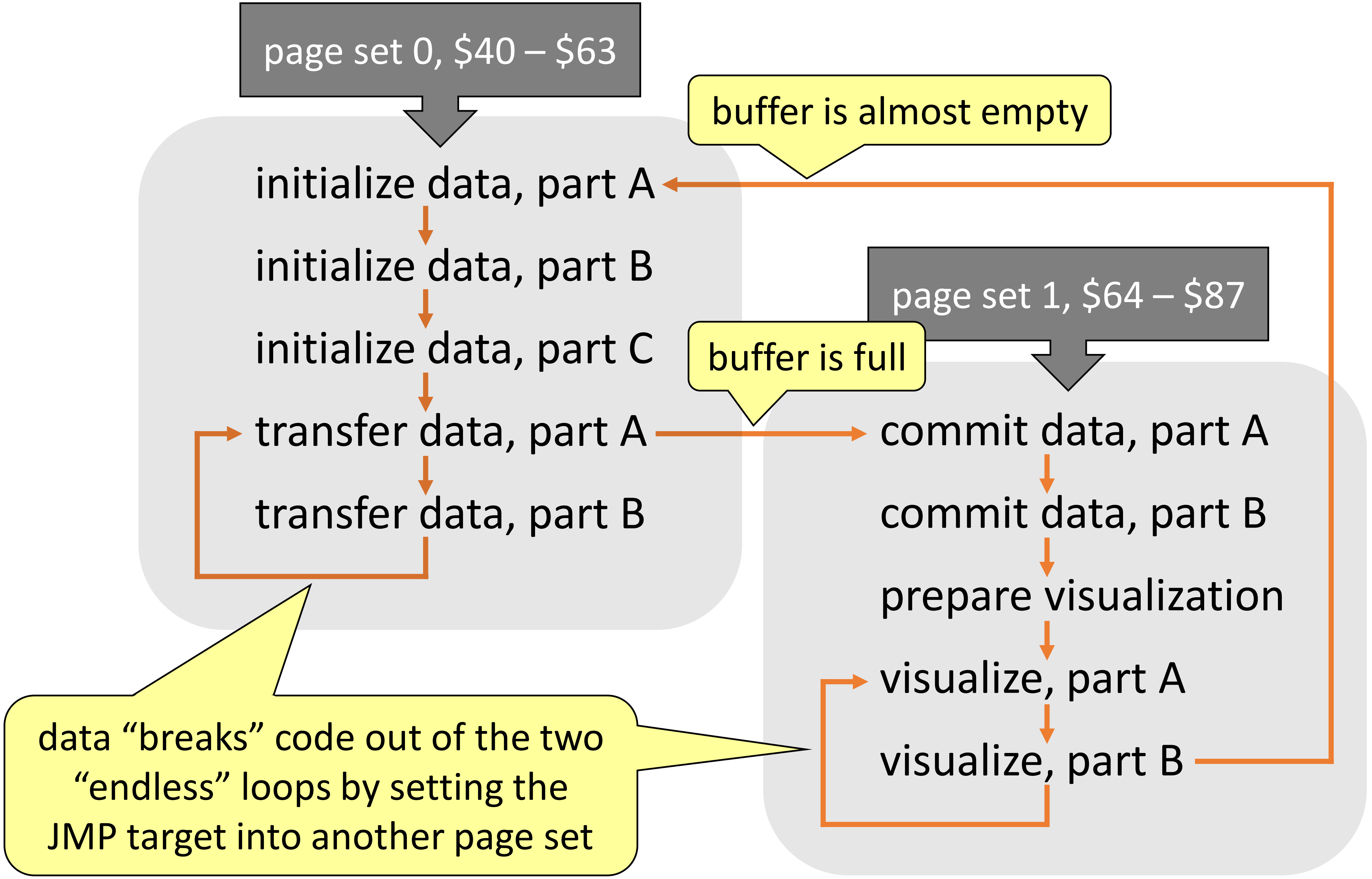
Is this bug? No, it’s a feature…

Loops


Initialize data block, part A
Initialize data block, part B
Initialize data block, part C
usually a loop variable check but A2Stream has no cycles
Transfer 3 bytes into the buffer, part A
Transfer 3 bytes into the buffer, part B
Commit data block, part A
Commit data block, part B

Loops


page set 0
Initialize data block, part A
Initialize data block, part B
Initialize data block, part C
usually a loop variable check but A2Stream has no cycles
Transfer 3 bytes into the buffer, part A
Transfer 3 bytes into the buffer, part B
page set 1
Commit data block, part A
Commit data block, part B
those two pulse generator types have the same offset!

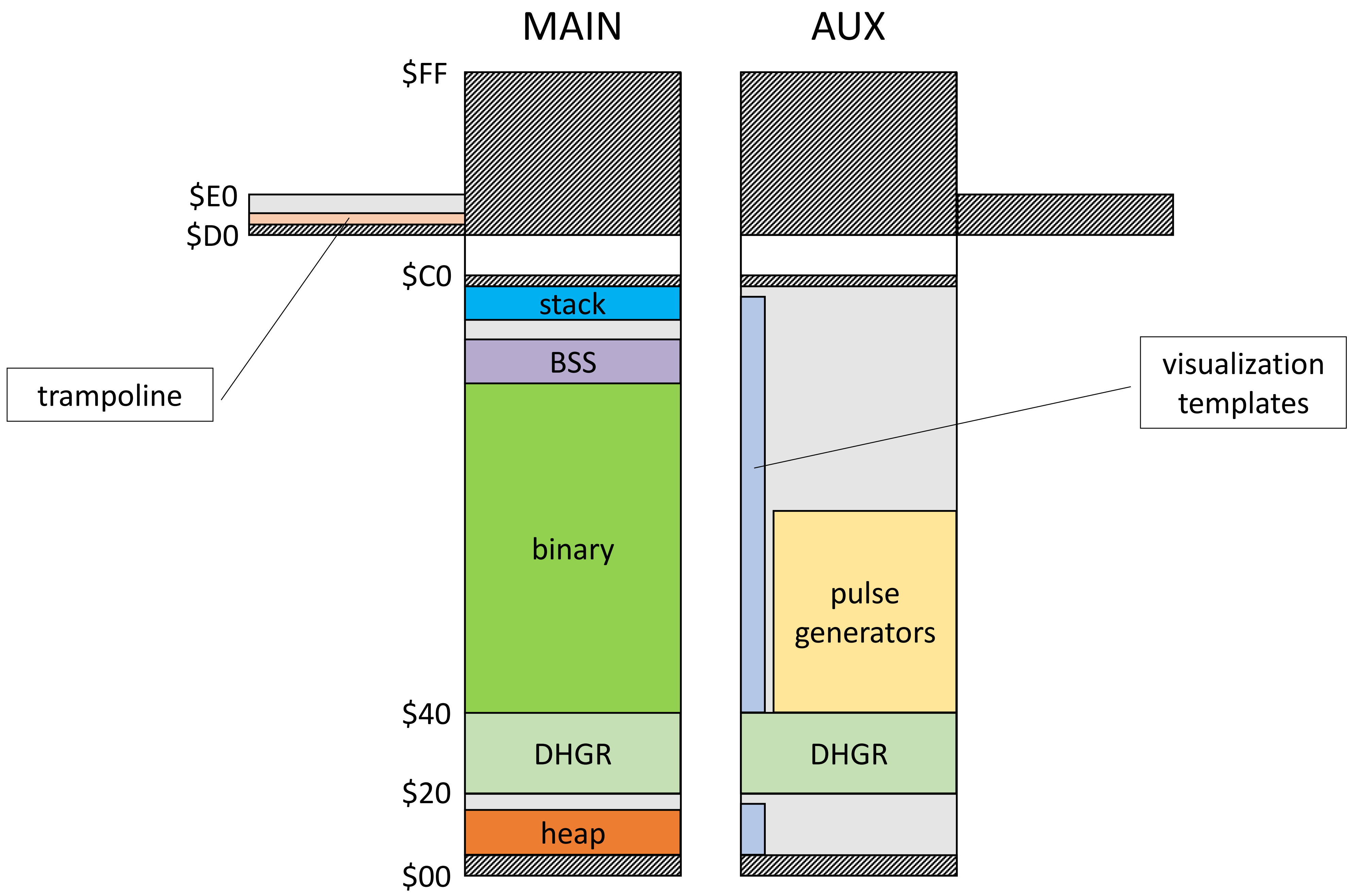

page set 0, $40 – $63
buffer is almost empty
initialize data, part A
initialize data, part B
initialize data, part C
transfer data, part A
transfer data, part B
page set 1, $64 – $87
buffer is full
commit data, part A
commit data, part B
prepare visualization
visualize, part A
visualize, part B
data “breaks” code out of the two “endless” loops by setting the JMP target into another page set

Visualization

- Completely precomputed
- 140 visualization templates
- Each template can be either of 2 types
  1. Odd bytes of a double hires line to be placed in MAIN memory
  2. Even bytes of a double hires line to be placed in AUX memory
- Every 256th data byte is a visualization value inserted into the stream
- One pulse generator type reads two bytes from the ring buffer
- The visualization byte selects one of the 140 templates
- The selected template is copied to the bottom 6 hires lines

Memory Layout

- Similar to the duty cycles, every template is on a different page
- 140 templates on the pages \$10 – \$1D and \$40 – \$BD
  - Pages \$40 – \$87 are shared with the pulse generators
- The A2Stream player runs completely in AUX memory
- No MAIN⇔AUX copy
- Double hires data and templates are loaded from W5100 into AUX
- Pulse generators are generated by C program in MAIN into AUX
  - Highly optimized C code places relevant variables in zero page, no ALTZP
- Trampoline in language card above ProDOS QUIT code


MAIN
AUX
$FF
$E0
$D0
$C0
$40
$20
$00
trampoline
stack
BSS
binary
DHGR
heap
visualization templates
pulse generators
DHGR

Q & A