

IC12A 74LS74

Pin 1: GND

Pin 2: ISA\_CLK

Pin 3: GND

Pin 4: VCC

Pin 5: SAA\_CLK

Pin 6: GND

Pin 7: GND

Pin 8: GND

Pin 9: GND

Pin 10: GND

Pin 11: GND

Pin 12: GND

Pin 13: GND

Pin 14: GND

Pin 15: GND

Pin 16: GND

Pin 17: GND

Pin 18: GND

Pin 19: GND

Pin 20: GND

Pin 21: GND

Pin 22: GND

Pin 23: GND

Pin 24: GND

Pin 25: GND

Pin 26: GND

Pin 27: GND

Pin 28: GND

Pin 29: GND

Pin 30: GND

Pin 31: GND

Pin 32: GND

Pin 33: GND

Pin 34: GND

Pin 35: GND

Pin 36: GND

Pin 37: GND

Pin 38: GND

Pin 39: GND

Pin 40: GND

Pin 41: GND

Pin 42: GND

Pin 43: GND

Pin 44: GND

Pin 45: GND

Pin 46: GND

Pin 47: GND

Pin 48: GND

Pin 49: GND

Pin 50: GND

Pin 51: GND

Pin 52: GND

Pin 53: GND

Pin 54: GND

Pin 55: GND

Pin 56: GND

Pin 57: GND

Pin 58: GND

Pin 59: GND

Pin 60: GND

Pin 61: GND

Pin 62: GND

Pin 63: GND

Pin 64: GND

Pin 65: GND

Pin 66: GND

Pin 67: GND

Pin 68: GND

Pin 69: GND

Pin 70: GND

Pin 71: GND

Pin 72: GND

Pin 73: GND

Pin 74: GND

Pin 75: GND

Pin 76: GND

Pin 77: GND

Pin 78: GND

Pin 79: GND

Pin 80: GND

Pin 81: GND

Pin 82: GND

Pin 83: GND

Pin 84: GND

Pin 85: GND

Pin 86: GND

Pin 87: GND

Pin 88: GND

Pin 89: GND

Pin 90: GND

Pin 91: GND

Pin 92: GND

Pin 93: GND

Pin 94: GND

Pin 95: GND

Pin 96: GND

Pin 97: GND

Pin 98: GND

Pin 99: GND

Pin 100: GND

Pin 101: GND

Pin 102: GND

Pin 103: GND

Pin 104: GND

Pin 105: GND

Pin 106: GND

Pin 107: GND

Pin 108: GND

Pin 109: GND

Pin 110: GND

Pin 111: GND

Pin 112: GND

Pin 113: GND

Pin 114: GND

Pin 115: GND

Pin 116: GND

Pin 117: GND

Pin 118: GND

Pin 119: GND

Pin 120: GND

Pin 121: GND

Pin 122: GND

Pin 123: GND

Pin 124: GND

Pin 125: GND

Pin 126: GND

Pin 127: GND

Pin 128: GND

Pin 129: GND

Pin 130: GND

Pin 131: GND

Pin 132: GND

Pin 133: GND

Pin 134: GND

Pin 135: GND

Pin 136: GND

Pin 137: GND

Pin 138: GND

Pin 139: GND

Pin 140: GND

Pin 141: GND

Pin 142: GND

Pin 143: GND

Pin 144: GND

Pin 145: GND

Pin 146: GND

Pin 147: GND

Pin 148: GND

Pin 149: GND

Pin 150: GND

Pin 151: GND

Pin 152: GND

Pin 153: GND

Pin 154: GND

Pin 155: GND

Pin 156: GND

Pin 157: GND

Pin 158: GND

Pin 159: GND

Pin 160: GND

Pin 161: GND

Pin 162: GND

Pin 163: GND

Pin 164: GND

Pin 165: GND

Pin 166: GND

Pin 167: GND

Pin 168: GND

Pin 169: GND

Pin 170: GND

Pin 171: GND

Pin 172: GND

Pin 173: GND

Pin 174: GND

Pin 175: GND

Pin 176: GND

Pin 177: GND

Pin 178: GND

Pin 179: GND

Pin 180: GND

Pin 181: GND

Pin 182: GND

Pin 183: GND

Pin 184: GND

Pin 185: GND

Pin 186: GND

Pin 187: GND

Pin 188: GND

Pin 189: GND

Pin 190: GND

Pin 191: GND

Pin 192: GND

Pin 193: GND

Pin 194: GND

Pin 195: GND

Pin 196: GND

Pin 197: GND

Pin 198: GND

Pin 199: GND

Pin 200: GND

Pin 201: GND

Pin 202: GND

Pin 203: GND

Pin 204: GND

Pin 205: GND

Pin 206: GND

Pin 207: GND

Pin 208: GND

Pin 209: GND

Pin 210: GND

Pin 211: GND

Pin 212: GND

Pin 213: GND

Pin 214: GND

Pin 215: GND

Pin 216: GND

Pin 217: GND

Pin 218: GND

Pin 219: GND

Pin 220: GND

Pin 221: GND

Pin 222: GND

Pin 223: GND

Pin 224: GND

Pin 225: GND

Pin 226: GND

Pin 227: GND

Pin 228: GND

Pin 229: GND

Pin 230: GND

Pin 231: GND

Pin 232: GND

Pin 233: GND

Pin 234: GND

Pin 235: GND

Pin 236: GND

Pin 237: GND

Pin 238: GND

Pin 239: GND

Pin 240: GND

Pin 241: GND

Pin 242: GND

Pin 243: GND

Pin 244: GND

Pin 245: GND

Pin 246: GND

Pin 247: GND

Pin 248: GND

Pin 249: GND

Pin 250: GND

Pin 251: GND

Pin 252: GND

Pin 253: GND

Pin 254: GND

Pin 255: GND

Pin 256: GND

Pin 257: GND

Pin 258: GND

Pin 259: GND

Pin 260: GND

Pin 261: GND

Pin 262: GND

Pin 263: GND

Pin 264: GND

Pin 265: GND

Pin 266: GND

Pin 267: GND

Pin 268: GND

Pin 269: GND

Pin 270: GND

Pin 271: GND

Pin 272: GND

Pin 273: GND

Pin 274: GND

Pin 275: GND

Pin 276: GND

Pin 277: GND

Pin 278: GND

Pin 279: GND

Pin 280: GND

Pin 281: GND

Pin 282: GND

Pin 283: GND

Pin 284: GND

Pin 285: GND

Pin 286: GND

Pin 287: GND

Pin 288: GND

Pin 289: GND

Pin 290: GND

Pin 291: GND

Pin 292: GND

Pin 293: GND

Pin 294: GND

Pin 295: GND

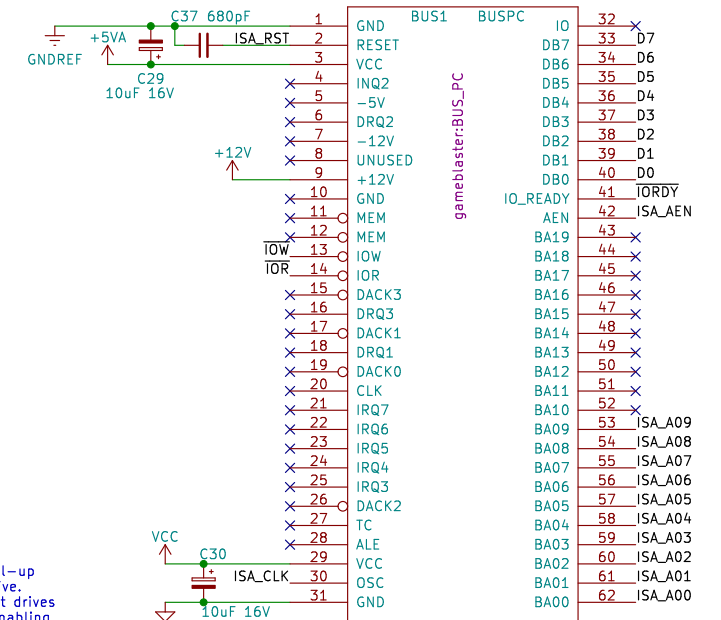
Pin 296: GND

ISA Bus clock is 14.318MHz, nominal SAA1099 clock is about 8MHz, the Game Blaster runs the SAA1099s at 7.159MHz by dividing the ISA clock in half.

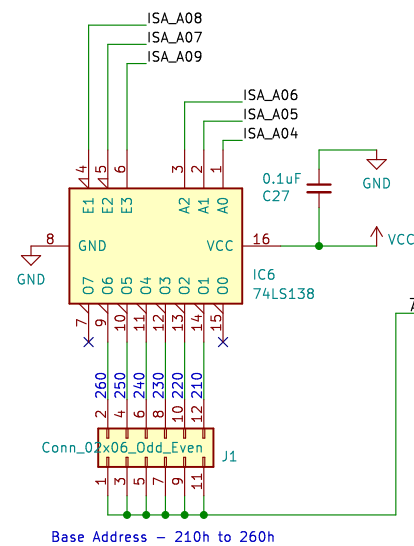
Bus transceivers are apparently good practice, so we take the ISA data bus D0-D7 and hook it up to a 74LS245 here. When IOR is pulled low (i.e. when we're reading from 2x4/2x8h or 2x4h on this card) we drive its direction pin to do B->A; otherwise, we're writing, and we run it A->B. CE is controlled by the output of the first decoder, which is low when an address within our base port range is accessed.

The circuit diagram illustrates a 2x8-bit parallel adder. It consists of two 74LS245 buffers (IC4 and IC5) and a 74LS374 8-bit register (RN1). The 74LS245 buffers are configured as tri-state buffers. The 74LS374 register is used to store the result of the addition. The circuit is powered by a 5V supply (VCC) and has a ground connection (GND). A 0.1μF capacitor (C28) is connected to the VCC pin of IC4. A 0.1μF capacitor (C20) is connected to the VCC pin of IC5. A 1K resistor (R6) is connected to the output of the 74LS374 register. The 74LS374 register is also connected to a 4.7K resistor (RN1). The 74LS245 buffers are connected to the 74LS374 register as follows: IC4 (A0-A7) is connected to the 74LS374 (D0-D7), and IC5 (A0-A7) is connected to the 74LS374 (D0-D7). The 74LS245 buffers are also connected to the 74LS374 register as follows: IC4 (B0-B7) is connected to the 74LS374 (D0-D7), and IC5 (B0-B7) is connected to the 74LS374 (D0-D7). The 74LS245 buffers are also connected to the 74LS374 register as follows: IC4 (B0-B7) is connected to the 74LS374 (D0-D7), and IC5 (B0-B7) is connected to the 74LS374 (D0-D7). The 74LS245 buffers are also connected to the 74LS374 register as follows: IC4 (B0-B7) is connected to the 74LS374 (D0-D7), and IC5 (B0-B7) is connected to the 74LS374 (D0-D7).

We use a 74LS374 octal flip-flop as a register to store the value written to  $2x6/2x7h$ , and read it back on  $2x4/2x8h$ , by putting DB0-DB7 on both the input and output.  $\overline{REG\_CLOCK}$  is the AND of  $IOW$  and  $2x6/2x7$ , which means that we latch data into the register when we write there.  $\overline{OE}$  is controlled by  $2xA/2xB$ , so that when we read from those ports, the outputs are placed onto the transceiver bus, and back through to the ISA bus. When the flip-flop outputs are inactive, DB0-DB6 are pulled up through 4.7K resistors to VCC, and DB7 is pulled down through a 1k resistor to ground, which puts  $0x7f$  on the bus if we read anywhere else (with the intent that we read it back on  $2x4h$ ).



## Port Range

[illegible]

This is only capable of decoding addresses in pairs, which is fine because the SAA1099s each have two registers. 2x0-2x1h are the left SAA1099, 2x2-2x3h are the right SAA1099. 2x4 is used on the original card to read back a status bit from the DIP-40, but conventional wisdom says "just read 7Fh". Card detection requires accessing a write to 2x6h (or 2x7h), and reading the same value back on 2xAh or 2xBh.

Base Address – 210h to 260h

The diagram shows two logic gates. IC9A (74LS08) is a 2-input AND gate. Its inputs are VCC (pin 1) and  $\overline{10W}$  (pin 2). Its output (pin 3) is connected to SAA10W through a pull-up resistor C35 (0.1uF) to GND. IC8A (74LS32) is a 2-input OR gate. Its inputs are  $\overline{10W}$  (pin 1) and  $2 \times 6 \text{ } 2 \times 7$  (pin 2). Its output (pin 3) is connected to REF\_CLOCK.

The original Game Blaster includes this AND of VCC and IOW to drive WR on the SAA1099. I suspect because the timing diagrams on the SAA1099 datasheet require that WR goes low slightly after CS goes low to start the data transfer. (Are these caps here to add delay? How do capacitors work?)

Since we want to save the contents of the bus into the detection register when a write is made to 2x6h or 2x7h, we clock the 74LS374 when 10W and 2x6-2x7 are both low. (i.e. when the host indicates a write, and when the address selected is the "ID write" port)

As long as the ISA bus isn't reset, the pull-up drives Pre-set on the flip flop high/inactive. When the SAA1099 finishes a read cycle, it drives DTACK low, which drives Pre-set active, enabling the buffer to pull IORDY low, finishing the wait-state.

The diagram illustrates the logic for generating the IORDY signal. The SAA1099's DTACK signal (pin 5) is pulled up to VCC by resistor R19 (4.7K). It is connected to the input of IC10B (74LS125). The output of IC10B (pin 6) is connected to the D input of IC12B (74LS125). The SAA1099's ISA\_RST signal (pin 4) is connected to the input of IC10A (74LS125). The output of IC10A (pin 3) is connected to the clock input (pin 10) of IC12B. The SAA1099's 2x0\_2x1 signal (pin 12) and 2x2\_2x3 signal (pin 13) are connected to the inputs of IC9D (74LS08). The output of IC9D (pin 11) is connected to the D input of IC12B. The SAA1099's VCC (pin 2) is connected to the input of IC10A. The output of IC10A (pin 3) is connected to the clock input (pin 10) of IC12B. The SAA1099's IOW signal (pin 1) is connected to the input of IC10A. The output of IC10A (pin 3) is connected to the clock input (pin 10) of IC12B. The SAA1099's VCC (pin 10) is connected to the input of IC12B. The output of IC12B (pin 9) is connected to the input of IC10C (74LS125). The output of IC10C (pin 8) is connected to the IORDY signal (pin 8). The SAA1099's GND (pin 9) is connected to the input of IC10C. The output of IC10C (pin 8) is connected to the IORDY signal (pin 8). The SAA1099's VCC (pin 10) is connected to the input of IC12B. The output of IC12B (pin 9) is connected to the input of IC10C. The output of IC10C (pin 8) is connected to the IORDY signal (pin 8). The SAA1099's GND (pin 9) is connected to the input of IC10C. The output of IC10C (pin 8) is connected to the IORDY signal (pin 8).

As long as the ISA bus isn't reset, the pull-up drives Pre-set on the flip flop high/inactive. When the SAA1099 finishes a read cycle, it drives DTACK low, which drives Pre-set active, enabling the buffer to pull IORDY low, finishing the wait-state.

2x0h through 2x3h are  $\overline{CS}$  for the SAA1099s; selecting either puts a 0 onto Data for the flip flop, and writing to it clocks the data to the buffer. When a write happens, the buffer control pin goes low, pulling IO\_CH\_READY on the ISA bus low, introducing a wait state.

(Leave RP1 unpopulated and replace R1 and R2 with 4.7K resistors for a fixed "medium" volume)

Hey, look! It's a 23.417KHz RC low-pass filter!

Aww yeah we coupling

LEFT\_RAW  
RIGHT\_RAW  
VDDA  
GNDREF  
LEFT\_IN  
RIGHT\_IN

R13 1.5K  
R11 10K  
R10 1.5K  
R12 10K  
R1 15K  
R2 15K  
C24 680pF  
C36 680pF  
C26 680pF  
C25 680pF  
C17 1uF tant  
C23 1uF tant  
RP1A 50K  
RP1B 50K

(Leave RP1 unpopulated and replace R1 and R2 with 4.7K resistors for a fixed "medium" volume)

RIGHT\_IN

LEFT\_IN

+5V is low for a TEA2025B, but apparently still in operating range. (The Sound Blaster used a 12V -> 9V regulator, but the Game Blaster just used this zener-regulated 12V -> 5V circuit for all the analog power)

VDDA

R7 10K

GNDREF

0.1uF C19

+5VA

0.1uF C18

VDDA

R5 10K

6

18

2x0\_2x1 2

SAA\_CLK 8

SAA\_IOW 1

ISA\_A00\_3

DB0 10

DB1 11

DB2 12

DB3 13

DB4 14

DB5 15

DB6 16

DB7 17

IC2 SAA1099

4 RIGHT\_RAW

5 LEFT\_RAW

7 DTACK

9 VSS

2x2\_2x3 2

SAA\_CLK 8

SAA\_IOW 1

ISA\_A00\_3

DB0 10

DB1 11

DB2 12

DB3 13

DB4 14

DB5 15

DB6 16

DB7 17

IC3 SAA1099

4 RIGHT\_RAW

5 LEFT\_RAW

7 DTACK

9 VSS

- CS goes low ->  
- register 0/1 (cpu)

Because  $\overline{CS}$  is already guaranteeing we're accessing this specific SAA1099, we just use the A0 line of the ISA bus directly to determine which register to write to.

- $\overline{CS}$  goes low  $\rightarrow \overline{WR}$  goes low
- register 0/1 (control/data) selected by A0
- SAA1099 latches data from D0-D7 into register
- SAA1099 pulls  $\overline{DACK}$  low when data transfer is complete

The original Game Blaster uses a tantalum for the polarized cap here, and has a couple extra 0.1uF caps in the circuit. This circuit is lifted from the Sound Blaster.

The original Game Blaster uses a tantalum for the polarized cap here, and has a couple extra 0.1uF caps in the circuit. This circuit is lifted from the Sound Blaster.

The circuit diagram shows a 6-bit ripple carry adder implemented using six integrated circuits:

- IC8B (74LS32)**: A hex invertor used to invert input B (pin 4) to produce  $\bar{B}$  (pin 6).
- IC9B (74LS08)**: A 2-input AND gate that takes inputs A (pin 4) and  $\bar{B}$  (pin 5) to produce the propagate signal  $P = A \oplus B$  (pin 6).
- IC9C (74LS08)**: A 2-input AND gate that takes inputs A (pin 9) and B (pin 10) to produce the generate signal  $G = AB$  (pin 8).
- IC10D (74LS125)**: A 3-state buffer that takes input A (pin 12) and produces output A (pin 11). Its enable pin (pin 10) is connected to GND.
- IC8C (74LS32)**: A hex invertor used to invert the propagate signal P (pin 9) to produce  $\bar{P}$  (pin 8).
- IC8D (74LS32)**: A hex invertor used to invert input B (pin 12) to produce  $\bar{B}$  (pin 13).

The final sum outputs are produced by two more 2-input AND gates (not explicitly labeled but implied by the connections):

- Sum bit 0 ( $S_0$ )**: Produced by an AND gate taking inputs A (pin 4) and  $\bar{B}$  (pin 5) from IC8B, resulting in  $A \oplus B$ .
- Sum bit 1 ( $S_1$ )**: Produced by an AND gate taking inputs A (pin 9) and  $\bar{B}$  (pin 13) from IC9C and IC8D, resulting in  $A \oplus B$ .
- Sum bit 2 ( $S_2$ )**: Produced by an AND gate taking inputs A (pin 4) and B (pin 10) from IC9B, resulting in  $AB$ .
- Sum bit 3 ( $S_3$ )**: Produced by an AND gate taking inputs A (pin 9) and  $\bar{B}$  (pin 8) from IC9C and IC8C, resulting in  $A \oplus B$ .
- Sum bit 4 ( $S_4$ )**: Produced by an AND gate taking inputs A (pin 4) and  $\bar{B}$  (pin 5) from IC8B, resulting in  $A \oplus B$ .
- Sum bit 5 ( $S_5$ )**: Produced by an AND gate taking inputs A (pin 9) and B (pin 10) from IC9C, resulting in  $AB$ .

The carry-in (Cin) is connected to GND (pin 1), and the carry-out (Cout) is connected to VCC (pin 14).

**Derivative Labs, Inc.**

Sheet: /  
File: gameblaster.kicad\_sch

**Title: CT-1300C**

Size: A3	Date: 2023-02-07
----------	------------------

SIZE: 100 MB	DATE: 2023-01-10
KiCad E.D.A.	kicad (6.0.11)

Rev: 1

id: 1/1