Token-ring with EMMA II

For this exercise, you will need two or more computers (EMMA II). The computers should be connected in series in a ring, similar to the old token-ring networks. Data is sent from computer to computer in serial form. Each computer has an 8 bit shift register through which the bits are switched. The ring of computers then forms a long circular shift register. A computer in the ring acts as a master and the others as slaves. The master determines the rate of data transmission through a separate clock signal. I / O MONITOR plates connected to each computer show the contents of the shift registers with LEDs.

1. Connect the D0-D7 inputs on the I / O MONITOR plates to the PB0-PB7 outputs on each computer. Check that the I / O MONITOR plates are in READ mode.

2. Connect PA0 on the master to PA0 on each slave. This is the clock signal that determines the rate of data transfer.

3. Connect PA2 to PA1 on the next computer so that the computers form a ring. Here is the data transfer itself.

4. Connect 0V and + 5V to all computers and I / O MONITOR plates.

5. Double-check the connections and turn on the power.

6. Enter the master program in the master and slave the program in each slave. Double check that the programs are properly entered.

7. Start each slave program first at address 0200. Then start the master program at the same address and enjoy!

8. Använd Use RESET on EMMA II to cancel the programs.

9. Try changing the speed of the master program by changing the value of address 0220.

10. Try changing the pattern that changes around by changing the value of address 021E.

11. Challenge: How many computers do you manage to connect to the ring?

Token-ring master

Adress	Kod	Assembler	Kommentar
0200	A9 05	LDA #05	; P_{A0} = Out (Clock), P_{A1} = In, P_{A2} = Out
0202	8D 03 09	STA 0903	
0205	A9 FF	LDA #FF	; P_{B0}-P_{B7} = Out (LEDs)
0207	8D 02 09	STA 0902	
020A	A9 01	LDA #01	; Set the clock signal high
020C	A2 00	LDX #00	
020E	8D 01 09	STA 0901	; Send clock signal on (P_{A0})
0211	49 01	EOR #01	; Inver the clock signal
0213	A0 01	LDY #01	; Short delay
0215	20 51 02	JSR 0251	
0218	E8	INX	
0219	E0 80	CPX #80	; Loop 128 times to empty
021B	D0 F1	BNE F1 (-15)	; shift registers in all slaves
021D	A2 01	LDX #09	; The shift register’s (X) starting value
021F	A0 60	LDY #60	; Long delay
0221	20 51 02	JSR 0251	
0224	8A	TXA	; Switch left and save bit 8 in 0020
0225	A2 00	LDX #00	
0227	0A	ASL	
0228	90 02	BCC 02 (+2)	
022A	A2 04	LDX #04	
022C	86 20	STX 20	
022E	AA	TAX	
022F	AD 01 09	LDA 0901	
0232	09 01	ORA #01	; Set the clock signal high (P_{A0})
0234	8D 01 09	STA 0901	
0237	29 02	AND #02	; Mask P_{A1} and save as bit0
0239	F0 01	BEQ 01 (+1)	; shift register
023B	E8	INX	
023C	8E 00 09	STX 0900	; Type the shift register to P_{B0}-P_{B7}
023F	A0 01	LDY #01	; Short delay
0241	20 51 02	JSR 0251	
0244	AD 01 09	LDA 0901	
0247	29 FA	AND #FA	; Set clock signal low (P_{A0})
0249	05 20	ORA 20	; Type bit 8 to P_{A2}
024B	8D 01 09	STA 0901	
024E	4C 1F 02	JMP 021F	; Loop
0251	48	PHA	; Delay routine, time is specified in Y
0252	8A	TXA	; A and X registers are saved
0253	48	PHA	
0254	A2 00	LDX #00	
0256	CA	DEX	
0257	D0 FD	BNE FD (-3)	
0259	88	DEY	
025A	D0 F8	BNE F8 (-8)	
025C	68	PLA	
025D	AA	TAX	
025E	68	PLA	
025F	60	RTS	

Token-ring slave

Adress	Kod	Assembler	Kommentar
0200	A9 04	LDA #04	; P_{A0} = in (Clock), P_{A1} = in, P_{A2} = out
0202	8D 03 09	STA 0903	
0205	A9 FF	LDA #FF	; P_{B0}-P_{B7} = out (LEDs)
0207	8D 02 09	STA 0902	
020A	A2 00	LDX #00	; Shift register’s starting value
020C	AD 01 09	LDA 0901	; Wait for the clock’s positive edge (P_{A0})
020F	29 01	AND #01	
0211	F0 F9	BEQ F9 (-7)	
0213	8A	TXA	; Switch left and save bit8 in 0020
0214	A2 00	LDX #00	
0216	0A	ASL	
0217	90 02	BCC 02 (+2)	
0219	A2 04	LDX #04	
021B	86 20	STX 20	
021D	AA	TAX	
021E	AD 01 09	LDA 0901	; Read PA1 and save as bit0 in
0221	29 02	AND #02	; shift register
0223	F0 01	BEQ 01 (+1)	
0225	E8	INX	
0226	8E 00 09	STX 0900	; Type the shift register to P_{B0}-P_{B7}
0229	AD 01 09	LDA 0901	; Wait for the negative edge of the clock
022C	29 01	AND #01	
022E	D0 F9	BNE F9 (-7)	
0230	AD 01 09	LDA 0901	; Type bit 8 to P_{A2}
0233	29 FA	AND #FA	
0235	05 20	ORA 20	
0237	8D 01 09	STA 0901	
023A	4C 0C 02	JMP 020C	; Loop