M2

[3] (10+6+4=20 marks) **Error Detection and Correction**

10 a) In the extended Hamming code, one creates $(n + k + 1)$-bit words. When reading the data, the check bits C and the overall parity P are calculated. State the different combinations of C and P. What do they mean? What actions should be followed in each case?

$C = C_8\, C_4\, C_2\, C_1$	P	Indication
0	0	No error (2)
$\neq 0$	1	Single error, can be corrected (3)
$\neq 0$	0	double error detected but cannot be corrected (3)
0	1	Error in P_{n+k+1} (Extra parity bit) (2)

6 b) The following combination of 8 bits of data and 4 bits of parity are read: 0 0 0 1 1 0 0 1 0 1 0 0. What was the original 8-bit data?

$P_1\, P_2\, D_3\, P_4$ $\quad$ 9 10 11 12

5 6 7 8

$C_1 = XOR(1, 3, 5, 7, 9, 11) = 1$ (1)

$C_2 = XOR(2, 3, 6, 7, 10, 11) = 1$ (1)

$C_4 = XOR(4, 5, 6, 7, 12) = 0$ (1)

$C_8 = XOR(8, 9, 10, 11, 12) = 0$ (1)

$C = C_8\, C_4\, C_2\, C_1 = 0\,0\,1\,1 \Rightarrow$ Error in bit 3. (1)

$P_1\ P_2\ \ P_4\ \ P_8$

Correct data: 0 0 1 1 1 0 0 1 0 1 0 0 (1) $= 1100\,D100$

$\underline{1}$ $\quad P_5\ P_6\ P_7\ \ D_9\ D_{10}\ D_{11}\ D_{12}$

D_3

4 c) Form the composite hamming code data-and-parity word for the data item 1001.

$n = 4 \Rightarrow k = 3$ (2) from $n \leq 2^k - k - 1$ (2)

P_1	P_2	D_3	P_4	D_5	D_6	D_7
0	0	1	1	0	0	1

(5)

$P_1 = XOR(3, 5, 7) = 0$ (1)

$P_2 = XOR(3, 6, 7) = 0$ (1)

$P_4 = XOR(5, 6, 7) = 1$ (1)

[4] (15 marks) **Computer Performance**

Consider two different implementations (i.e. machines) , M1 and M2, of the same instruction set. There are four classes of instructions (A, B, C, and D) in the instruction set. M1 has a clock rate of 500 MHz, M2 has a clock rate of 300 MHz. The average number of cycles for each instruction class on M1 and M2 is given in the following table.

Class	CPI on M1	CPI on M2	C1 usage	C2 usage	Third-part usage
A	4	2	30%	30%	50%
B	6	4	40%	20%	20%
C	8	3	20%	30%	20%
D	10	5	10%	20%	10%

The table also contains a summary of how three different compilers use the instruction set. C1 is a compiler produced by the makers of M1, C2 is a compiler produced by the makers of M2, and the other compiler is a third-party product. Assume that each compiler uses the same number of instructions for a given program but that the instruction mix is as described in the table.

- Using C1 on both M1 and M2, which machine is faster? How much faster is it?

EXE = CYC x TAU = INS x CPI / FRE

$EXE_{M1,C1} = INS \times (4\times0.3 + 6\times0.4 + 8\times0.2 + 10\times0.1)/500\times10^6 = 1.24\times10^{-8}\ INS$ s

$EXE_{M2,C1} = INS \times (2\times0.3 + 4\times0.4 + 3\times0.2 + 5\times0.1)/300\times10^6 = 2.1\times10^{-8}\ INS$ s

M2 is faster by $1.24/1.1 = 1.127$ about 13%

- Using C2 on both M2 and M1, which machine is faster? How much faster is it?

$EXE_{M1,C2} = INS \times (4\times0.3 + 6\times0.2 + 8\times0.3 + 10\times0.2)/500\times10^6 = 1.36\times10^{-8}\ INS$ s

$EXE_{M2,C2} = INS \times (2\times0.3 + 4\times0.2 + 3\times0.3 + 5\times0.2)/300\times10^6 = 1.1\times10^{-8}\ INS$ s

M2 is faster by $1.36/1.1 = 1.236$ about 24%

- If you purchase M1, which compiler would you use?

$EXE_{M1,C_3} = INS * (4\times0.5 + 6\times0.2 + 8\times0.2 + 10\times0.1)/500\times10^6 = 1.16\times10^{-8}\ INS$ s

~~C2~~ Compiler C_3

- If you purchase M2, which compiler would you use?

$EXE_{M2,C3} = INS * (2\times0.5 + 4\times0.2 + 3\times0.2 + 5\times0.1)/300\times10^6 = 0.96\times10^{-8}\ INS$ s

Third party compiler.

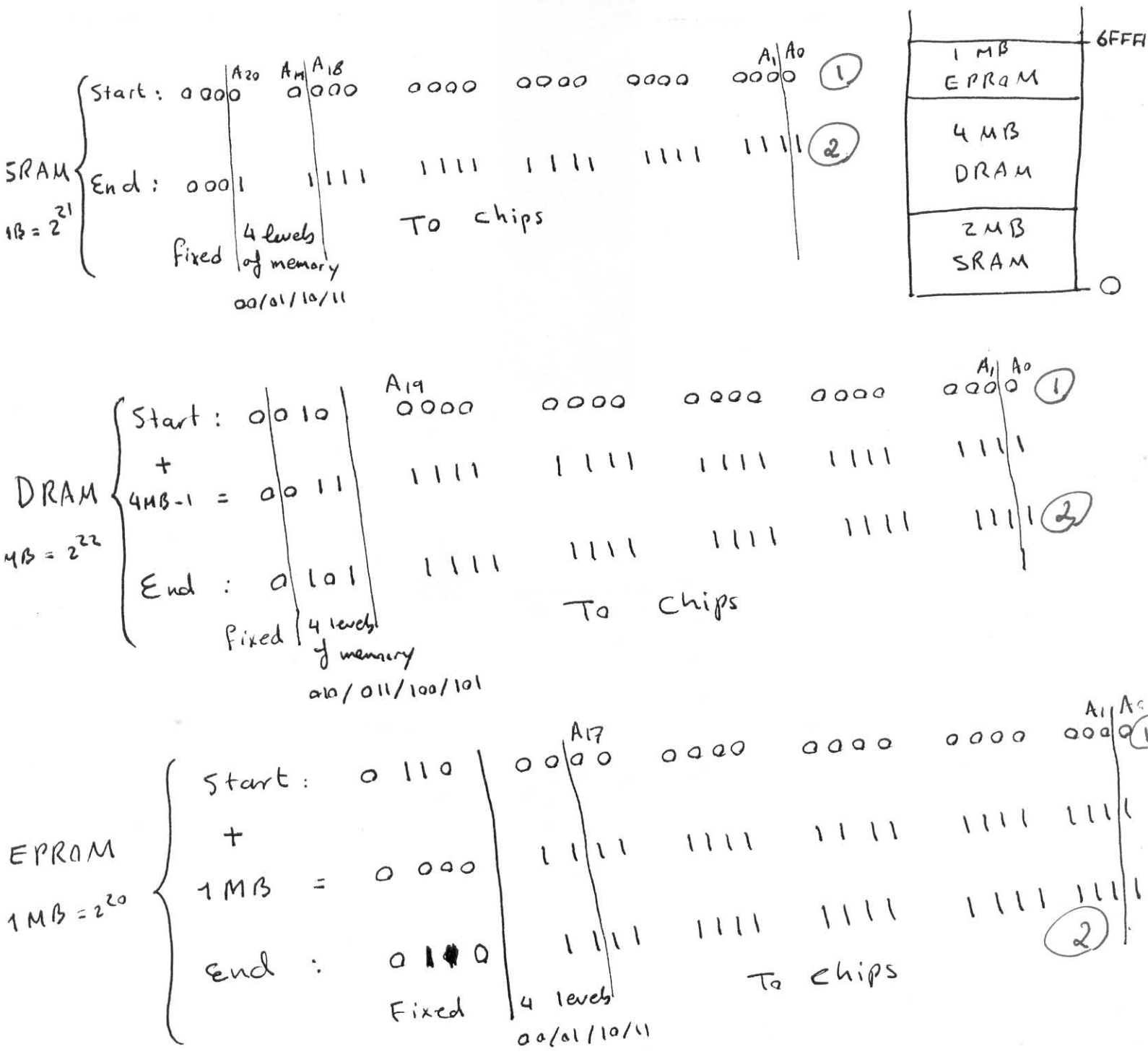
- Which machine would you purchase for maximum performance, if we assume that all other criteria are identical, including cost?

M2 with third party Compiler

MIPS = #ins / EXE x

[5] (25 marks) **Memory Design**

A computer system with an 80386SX microprocessor (24-bit address bus and 16-bit data bus) requires 7 MB of memory starting at (the bottom) address 0_H. This 7 MB should be divided into 2 MB of SRAM at the bottom, 4 MB of DRAM at the middle, and 1 MB of EPROM at the top. The available parts are 128K×16 EPROM chips, 256K×8 SRAM chips, 512K×4 DRAM chips, and PAL chips with 10 inputs and 8 outputs. The processor provides $\overline{\text{MRDC}}$, $\overline{\text{MWTC}}$, $\overline{\text{BHE}}$, and $\overline{\text{BLE}}$ (i.e. A0) control signals. The SRAM and DRAM chips have $\overline{\text{WE}}$, $\overline{\text{OE}}$, and $\overline{\text{CE}}$ control inputs. The EPROM chips have $\overline{\text{OE}}$ and $\overline{\text{CE}}$ control inputs. Write the design expressions for the PAL devices, draw a simple circuit diagram, and label the wires in your design. Do all calculations in this page and draw the circuit in the next page.



Circuit Diagram for [5]

SRAM : 2MB / 256KB = 8 chips (in 2 Banks) (1)

DRAM : 4MB / 512K (0.5B) = 16 chips (in 2 Banks) (1)

EPRAM : 1MB / 128K (2B) = 4 chips (in 1 Bank) (1)

$\overline{HWE} = \overline{MWRC} \cdot \overline{BHE}$ (1/2)

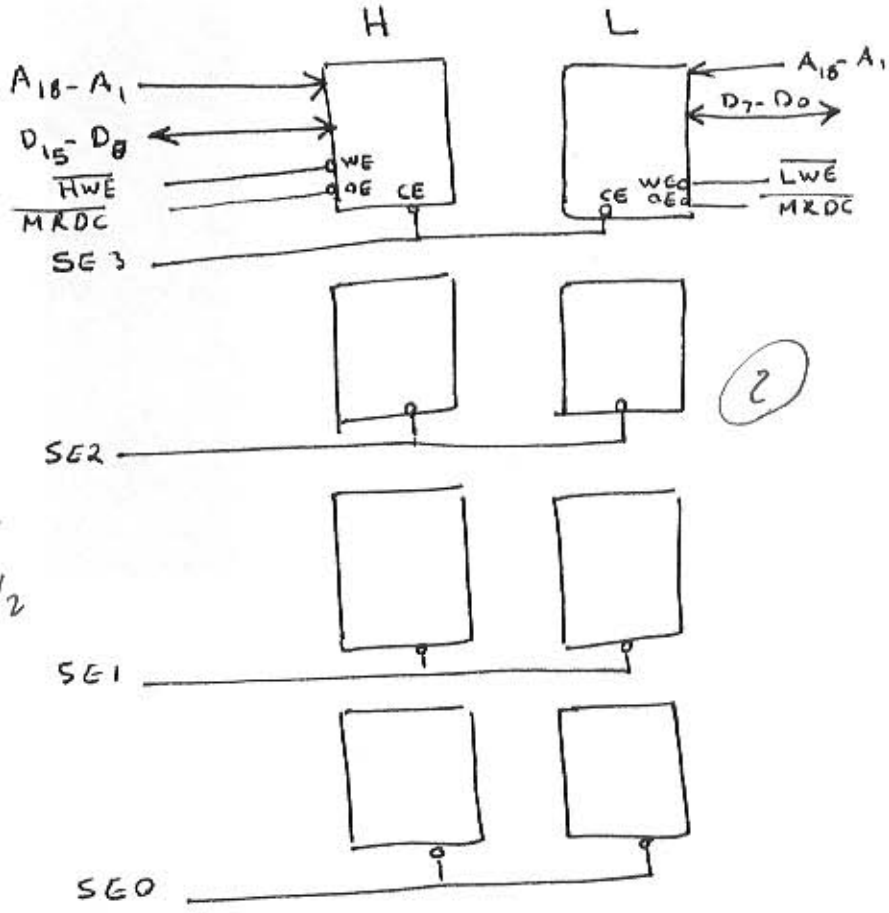
$\overline{LWE} = \overline{MWRC} \cdot \overline{BLE}$ (1/2)

$\overline{SE0} = \overline{A_{23}} \cdot \overline{A_{22}} \cdot \overline{A_{21}} \cdot \overline{A_{20}} \cdot \overline{A_{19}}$ (1/2)

$\overline{SE1} =$ " $\cdot \overline{A_{20}} \cdot A_{19}$ (1/2)

$\overline{SE2} =$ " $\cdot A_{20} \cdot \overline{A_{19}}$ (1/2)

$\overline{SE3} =$ " $\cdot A_{20} \cdot A_{19}$ (1/2)

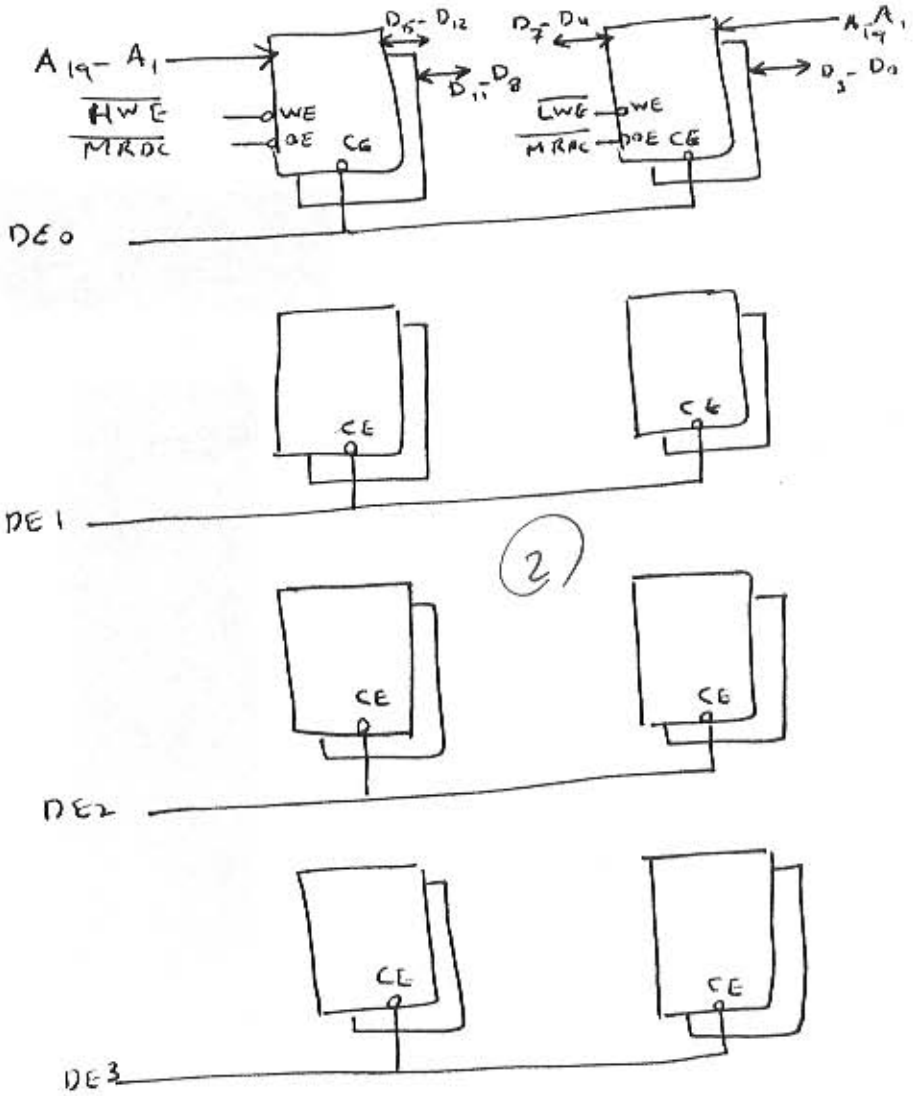


$\overline{DE0} = \overline{A_{23}} \cdot \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}}$

$\overline{DE1} = \overline{A_{23}} \cdot \overline{A_{22}} \cdot A_{21} \cdot A_{20}$

$\overline{DE2} = \overline{A_{23}} \cdot A_{22} \cdot \overline{A_{21}} \cdot \overline{A_{20}}$

$\overline{DE3} = \overline{A_{23}} \cdot A_{22} \cdot \overline{A_{21}} \cdot A_{20}$

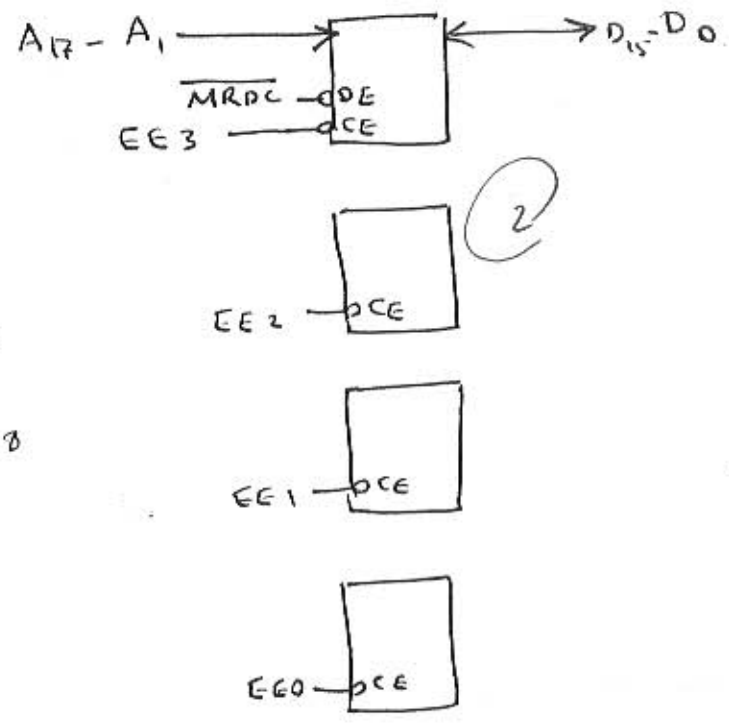


$\overline{EE0} = \overline{A_{23}} \cdot A_{22} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}}$

$\overline{EE1} = \overline{A_{23}} \cdot A_{22} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot A_{18}$

$\overline{EE2} = \overline{A_{23}} \cdot A_{22} \cdot A_{21} \cdot \overline{A_{20}} \cdot A_{19} \cdot \overline{A_{18}}$

$\overline{EE3} = \overline{A_{23}} \cdot A_{22} \cdot A_{21} \cdot \overline{A_{20}} \cdot A_{19} \cdot A_{18}$



[6] (15 marks) **Data Communication**

Compare the *maximum* bandwidth of a synchronous and an asynchronous bus. The synchronous bus has a clock cycle time of 10 ns, and each of its bus transmissions (to/from memory/IO) takes 1 clock cycle. The asynchronous bus requires 7 ns per handshake. The data bus in both cases is 32-bits wide. Find the bandwidth (in MB/s) for each bus when performing one double-word read from the memory and sending it to an IO device. The memory access time is 40 ns.

Synchronous Bus:

Step 1. Send address to memory: 10 ns (1)

Step 2. Read data: 40 ns (1)

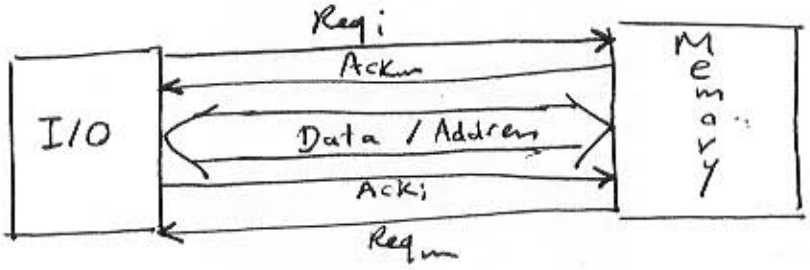
Step 3. Send data to device: 10 ns (1)

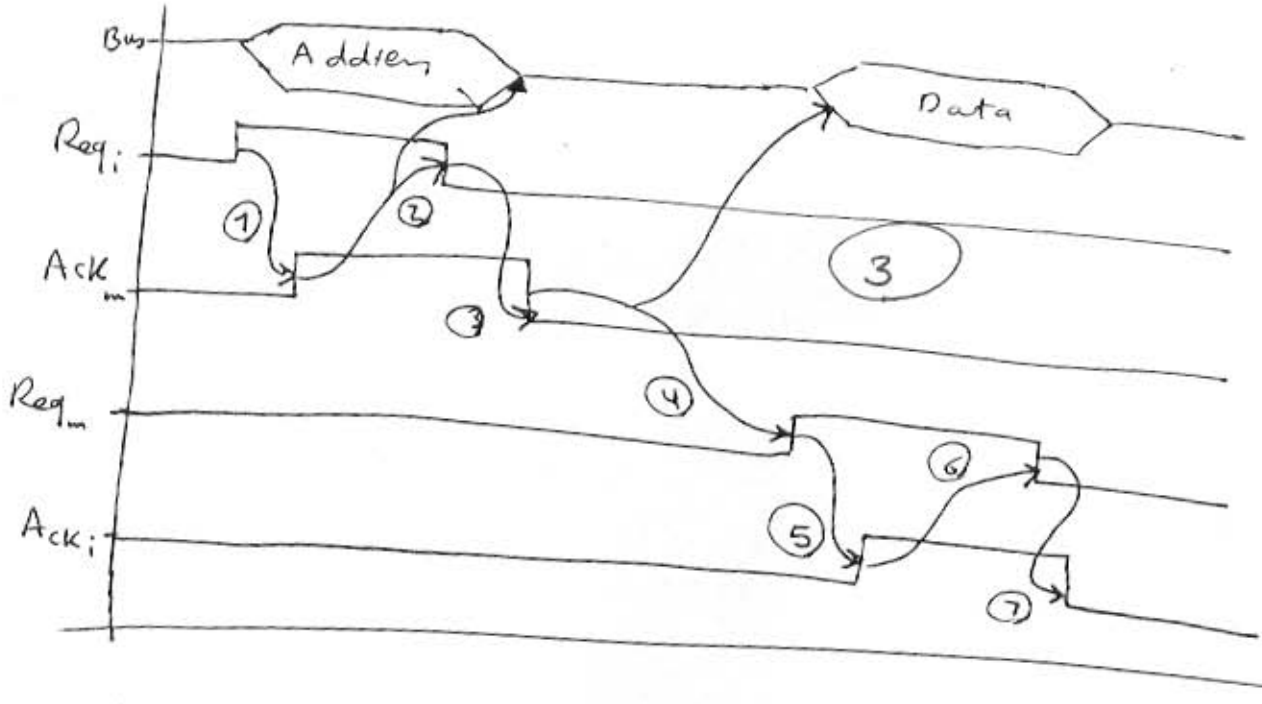
60 ns (1)

Bandwidth is 4 bytes every 60 ns.

$\frac{4 \text{ bytes}}{60 \text{ ns}} = 66.67 \text{ MB/sec}$ (1)

Asynchronous Bus:





There are 7 handshake signals.

2, 3, 84 can be done in parallel with memory time.

(1) Step 1. 7 ns

(3) Step 2, 3, 4 : Maximum of (3 x 7 ns, 40 ns) = 40 ns

(1) Steps 5, 6, 7 : 3 x 7 = 21 ns

(1) 68 ns

Bandwidth $\frac{4 \text{ bytes}}{68 \text{ ns}} = 58 \text{ MB/sec.}$ (1)

Synchronous bus is faster.