NTNU
Kunnskap for ei betre verd

Institutt for datateknikk og informasjonsvitenskap

Department of Computer and Information Science

# Examination paper for TDT4160 Computers and Digital Design

**Academic contact during examination: Gunnar Tufte**

**Phone: 97402478**

**Examination date: 11th of December 2017**

**Examination time (from-to):9:00 – 13:00**

**Permitted examination support material: D**

**Other information:**

**Language: English**

**Number of pages (front page excluded): 9**

**Number of pages enclosed: 3**

**Informasjon om trykking av eksamensoppgave**

**Originalen er:**

**1-sidig X** **2-sidig ☐**

**sort/hvit X** **farger ☐**

**skal ha flervalgskjema ☐**

**Checked by:**

_____

Date Signature

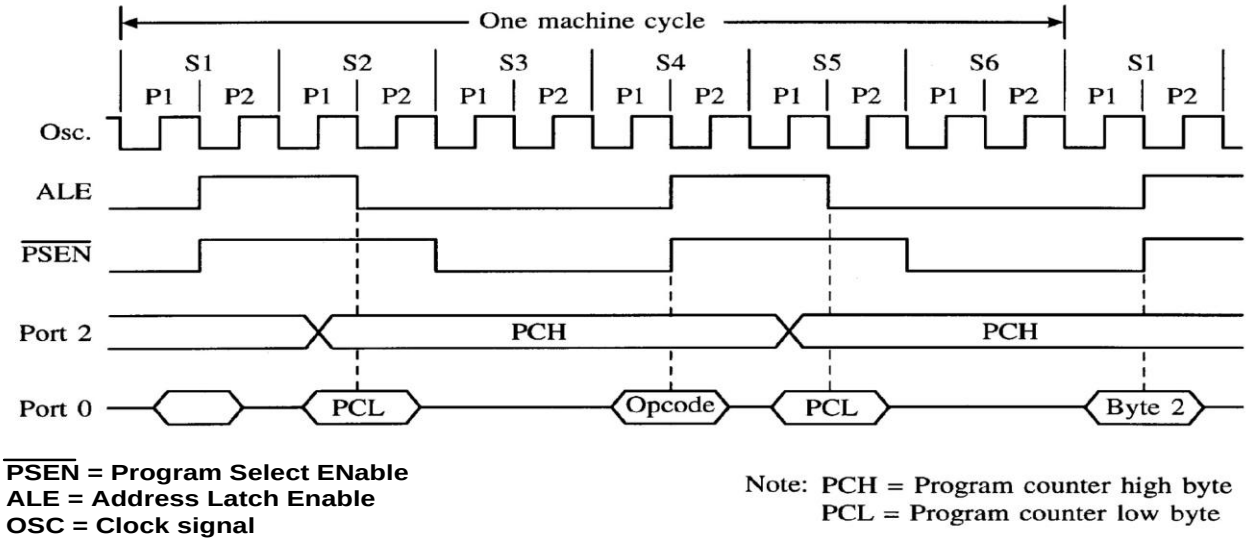
Merk! Studentane finn sensur i Studentweb. Har du spørsmål om sensuren må du kontakte instituttet ditt. Eksamenskontoret vil ikkje kunne svare på slike spørsmål.

## *Question 1 Start, miscellaneous (20% (a: 2.5%, b: 7.5%, c:.5%, d: 2.5% and e: 2.5%)*

a)
The term «stored program computer» was introduced together with the von Neumann architecture. What does the term mean?

b)
Figure 1 show a microcontroller's external memory interface. The bus signals in the figure illustrate reading of program memory. Port 0 and Port 2 are 8-bit ports. Answer the questions from the available information.

i) Is the bus transfer synchronous or asynchronous? Explain short.
ii) What is the size (maximum address space) for the program memory? Explain short.
iii) What is the number of bits in a word for this program memory? Explain short.




Figur 1 Bus interface.

c)
i) What does the term «superscalar» in processors mean?
ii) Is it possible to introduce superscalarity without influencing on the ISA?

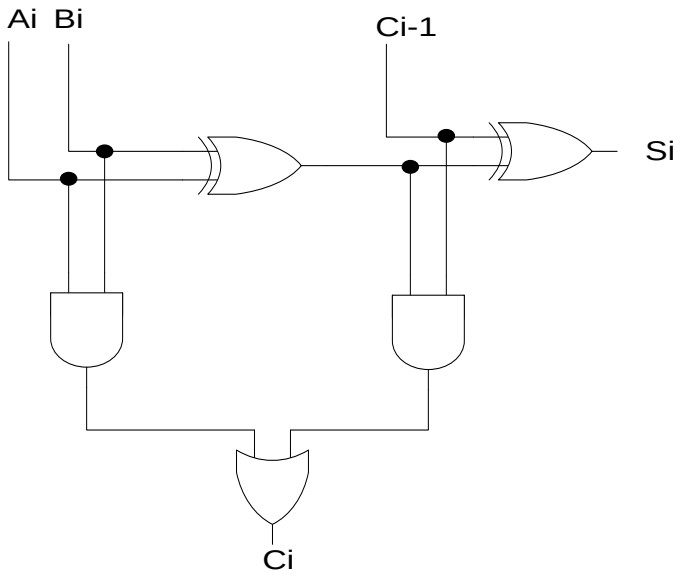
d)
What identify Harvard architecture in processors? Explain short

e)
«Latency hiding» is an important concept in modern computer architecture. Is cache part of this concept? Explain short why/why not.

### *Question 2 Digital Logic Level (20% (a: 4%, b: 7%, c: 7% and d: 2%))*

a)

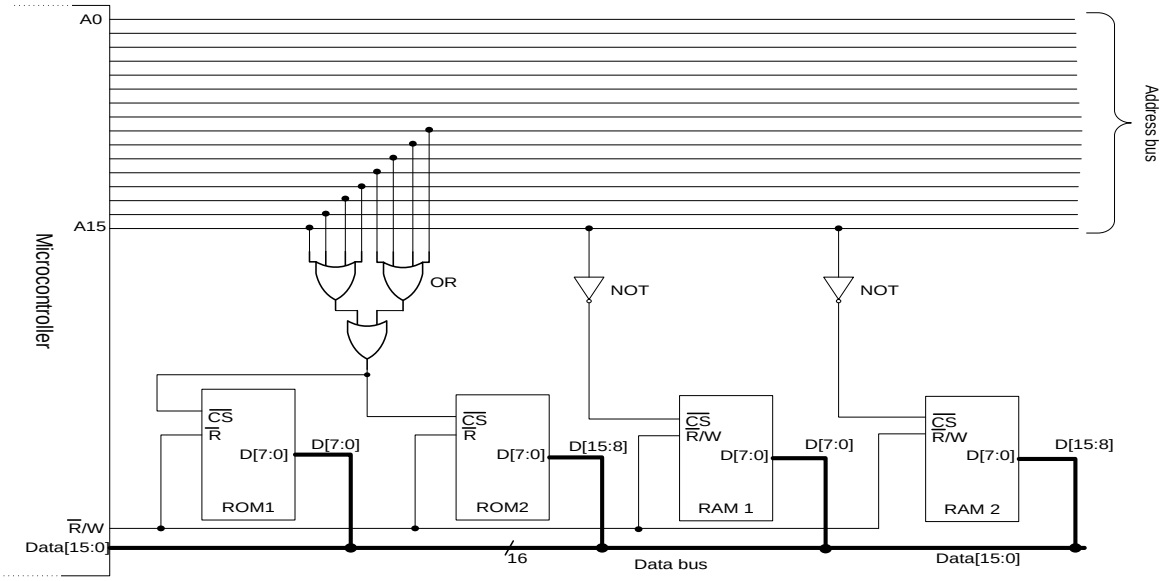
Figure 2 shows the logic for one of the functions in a 1-bit ALU. What function?




**Figur 2 Logic for 1-bit ALU-function.**

b)

An embedded system uses a 16-bit microcontroller. Figure 3 shows the external bus interface for the microcontroller with address decoding. There are two ROM chips used for program, and two RAM chips. All units on the bus uses an active low (logic "0") Chip Select (CS) signal.

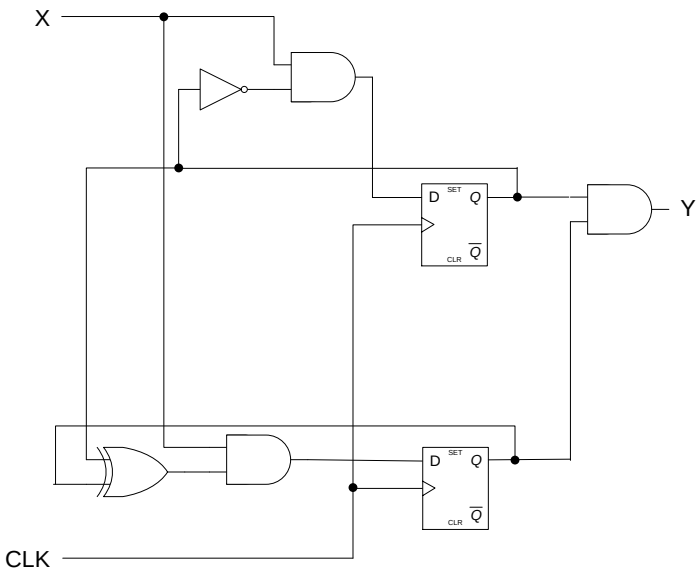



**Figur 3 Address decoding.**

i) Find the address space for the RAM and ROM.Or optionally free space and overlaps.

ii) Draw the memory map for the system

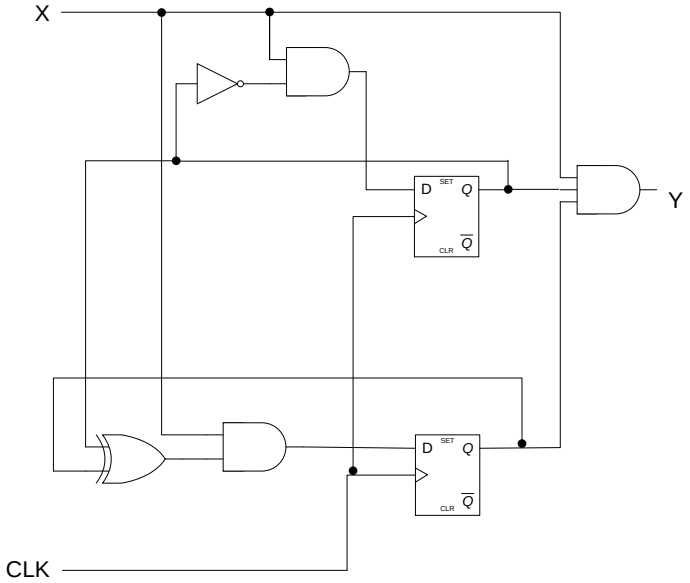
c)
Figure 4 shows a Finite State Machine (FSM). The D0 Q0 data flip-flop is at the top. The D1 Q1 data flip-flop is at the bottom.




Figur 4 Finite State Machine (FSM) 1.

Find the excitation equation for D0 and D1. Find the next state equations and the next-state-table for the circuit to show the behaviour of this FSM and the output signal Y.

d)
The FSM in Figure 4 is changed to the shown FSM in Figure 5. How does the changes influence on the behaviour of the output Y?




Figur 5 Finite State Machine (FSM) 2.

### *Question 3 Micro Architecture and micro instructions (20% (a: 5%, b: 5% and c: 10%))*

Use Figure 9, Figure 10, Figure 11 and Figure 12 for the IJVM in the appendix to answer the questions

a)
The device 4-to-16 Decoder in Figure 9 is used to control data on the B-bus. The C-Bus uses separate bit to control each register. Why?

b)
For the micro architecture in Figure 9. Make micro instruction(s) that execute a logical NOR. Data are in the TOS register and the LV register (*TOS NOR LV).* The result is to be stored in the H register and the MDR register.

You don't need to consider the Addr and J fields. Give the correct bit value for the ALU, C, Mem and B fields, see Figure 10.

c)
For the micro architecture in Figure 9. During execution of an instruction the value of MPC is 0xA3 (MPC points to address 0xA3 in control store).

MDR register is set to the following value: 0xAAAA AAAA.
H register set to the following value: 0x5555 5555.
LV register is set to the following: 0xA5A5 0000.

The table below show two subsets of the content of control store.
For the rest of the execution of this instruction. Explain shortly what each micro instruction do.
What is the value of the MDR register when MPC is set to the value 0x00?

| Control store Address | Next_Adr | JMPC | JAMN | JAMZ | SLL8 | SRA1 | F0 | F1 | ENA | ENB | INVA | INC | H | OPC | TOS | CPP | LV | SP | MDR | MAR | Write | Read | FETCH | B Buss | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0A3 | 010100100 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0A4 | 010100101 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0A5 | 010100110 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0A6 | 000000000 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | | | | | | | | | | | | : | | | | | | | | | | | | | | |
| 1A5 | 110100110 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1A6 | 110100111 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1A7 | 000000000 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1A8 | 110101001 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| | | | | | | | | | | | | | : | | | | | | | | | | | | | | |

### *Question 4 Instruction Set Architecture (ISA) (20% (a: 2.5%, b: 2.5%, c: 10 and d: 5%))*

RaM 007 is a very simple processor. RaM 007 includes a load and a store instruction, 8 ALU instructions, some special instruction including the NOP instruction and three flow control instructions. The formats of the instructions are shown in Figure 6. Figure 7 shows the instruction set. All registers and busses are 32 bit. There are 32 general registers available. The processor is a Harvard architecture type. Use Figure 6 and Figure 7 to solve the questions

a)
Does RaM 007 use fixed length instructions?

b)
Give an example of one instruction using «immediate addressing».

c)
R0 is set to the following value: 0x0000 FFFF, R8 is set to the following value: 0xFFFF 0000, R11 is set to the following value: 0x0000 0000, R12 is set to the following value: 0x0000 0003 and R13 is set to the following value: 0x0001 0005. In the data memory the following data is placed from address 0xFFFF 0000:

| Address | Data |
|---|---|
| 0xFFFF 0000: | 0x00 00 00 01 |
| 0xFFFF 0001: | 0x00 00 00 02 |
| 0xFFFF 0002: | 0x00 00 00 03 |
| 0xFFFF 0003: | 0x00 00 00 04 |
| 0xFFFF 0004: | 0x00 00 00 00 |
| 0xFFFF 0005: | 0x00 00 00 05 |
| 0xFFFF 0007: | 0x00 00 00 06 |
| 0xFFFF 0008: | 0x00 00 00 07 |
| 0xFFFF 0009: | 0x00 00 00 08 |

The following pseudo code is part of a larger program. The execution starts at program memory address 0000 FFFF. Answer the questions from the available information.

```
0x0000 FFFF: LOAD R1, R8;
0x0001 0000 ADD R28, R1, R1;
0x0001 0001: BZ R13;
0x0001 0002: ADD R11, R1, R11;
0x0001 0003: INC R8, R8;
0x0001 0004: BNZ R0;
0x0001 0005: STORE R11, R8;                :
```

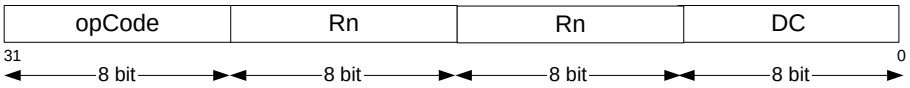
Explain the code. What value will R11 contain at the end of the shown code?

e)
In another run all register values and memory content are restored except R8 that now is set to the value: 0xFFFF 0004. What value will R11 contain at the end of this run of the shown code?

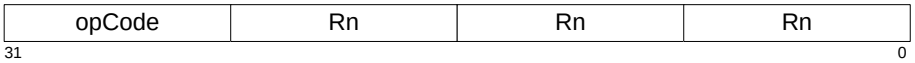
Load/store:

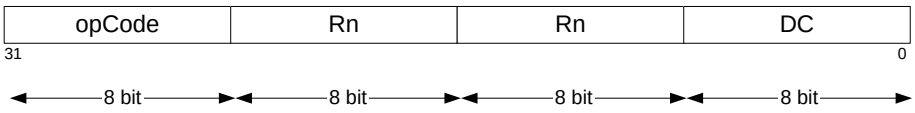
Load:


Store:



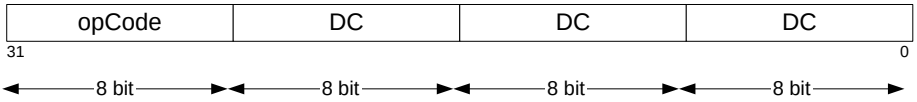

ALU:



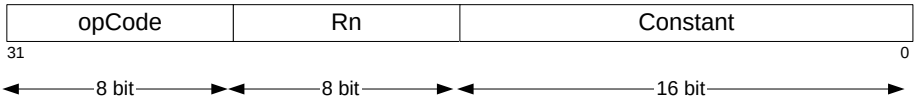




Spesial:

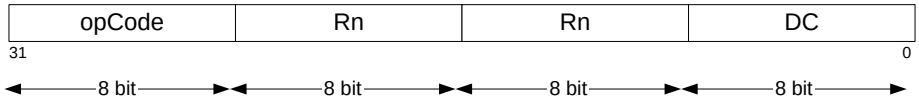
NOP:




MOVC:

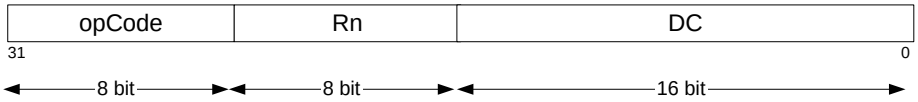



CP:

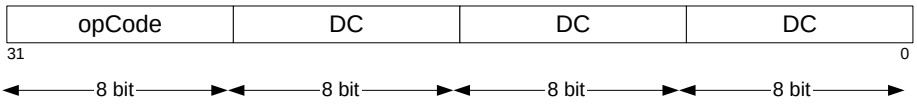



Flow control:

BZ/BNZ




RT




Rn: any user register, R0 - R31

DC: Don’t care: any data memory location

**Figur 6 Instruction format Ram 007**

**Instructions set:**

**LOAD:** Load data from memory.

**load Ri, Rj** Load register Ri from memory location in Rj.

**STORE:** Store data in memory.

**store Ri, Rj** Store register Ri in memory location in Rj.

**ALU:** Data manipulation, register–register operations.

**ADD Ri, Rj, Rk** ADD, $Ri = Rj + Rk$. Set Z-flag if result =0.
**NAND Ri, Rj, Rk** Bitwise NAND, $Ri = \overline{Rj \cdot Rk}$. Set Z-flag if result =0.
**OR Ri, Rj, Rk** Bitwice OR, $Ri = Rj + Rk$. Set Z-flag if result =0.
**INV Ri, Rj** Bitwice invert, $Ri = \overline{Rj}$. Set Z-flag if result =0.
**INC Ri, Rj** Increment, $Ri = Rj + 1$. Set Z-flag if result =0.
**DEC Ri, Rj** Decrement, $Ri = Rj - 1$. Set Z-flag if result =0.
**MUL Ri, Rj, Rk** Multiplication, $Ri = Rj * Rk$. Set Z-flag if result =0.
**CMP, Ri, Rj** Compare, Set Z-flag if $Ri = Rj$

**Special:** Misc.

**CP Ri, Rj** Copy, $Ri < -Rj$ (copy Rj into Ri)
**NOP** Waste of time, 1 clk cycle.
**MOVC Ri, constant** Put a constant in register $Ri = C$.

**Flow control:** Branch.

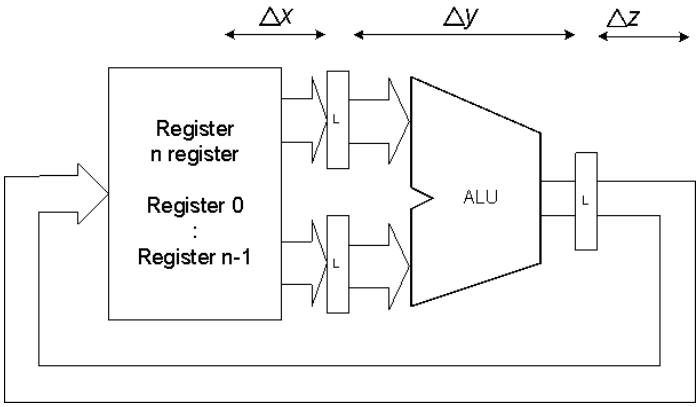
**BZ, Ri** Conditional branch on zero (Z-flag = 1) , $PC = Ri$.
**BNZ, Ri** Conditional branch on non zero (Z-flag = 0), $PC = Ri$.
**RT** Return, return from branch.

Ri, Rj and Rk: Any user register.
DC: Don't care.

Figur 7 Instruction set RaM 007.

## *Question 5 Performance ((20% (a: 5%, b: 10%, c: 2.5% and d: 2.5%))*

a)
Figure 8 shows a data path including a register bank, an ALU and three latches. Δx = 10ns, Δy = 25ns og ΔZ = 12ns. From the available information. What parameter limit the clock frequency? What is the maximum clock frequency? Explain short.




**Figur 8 Data path.**

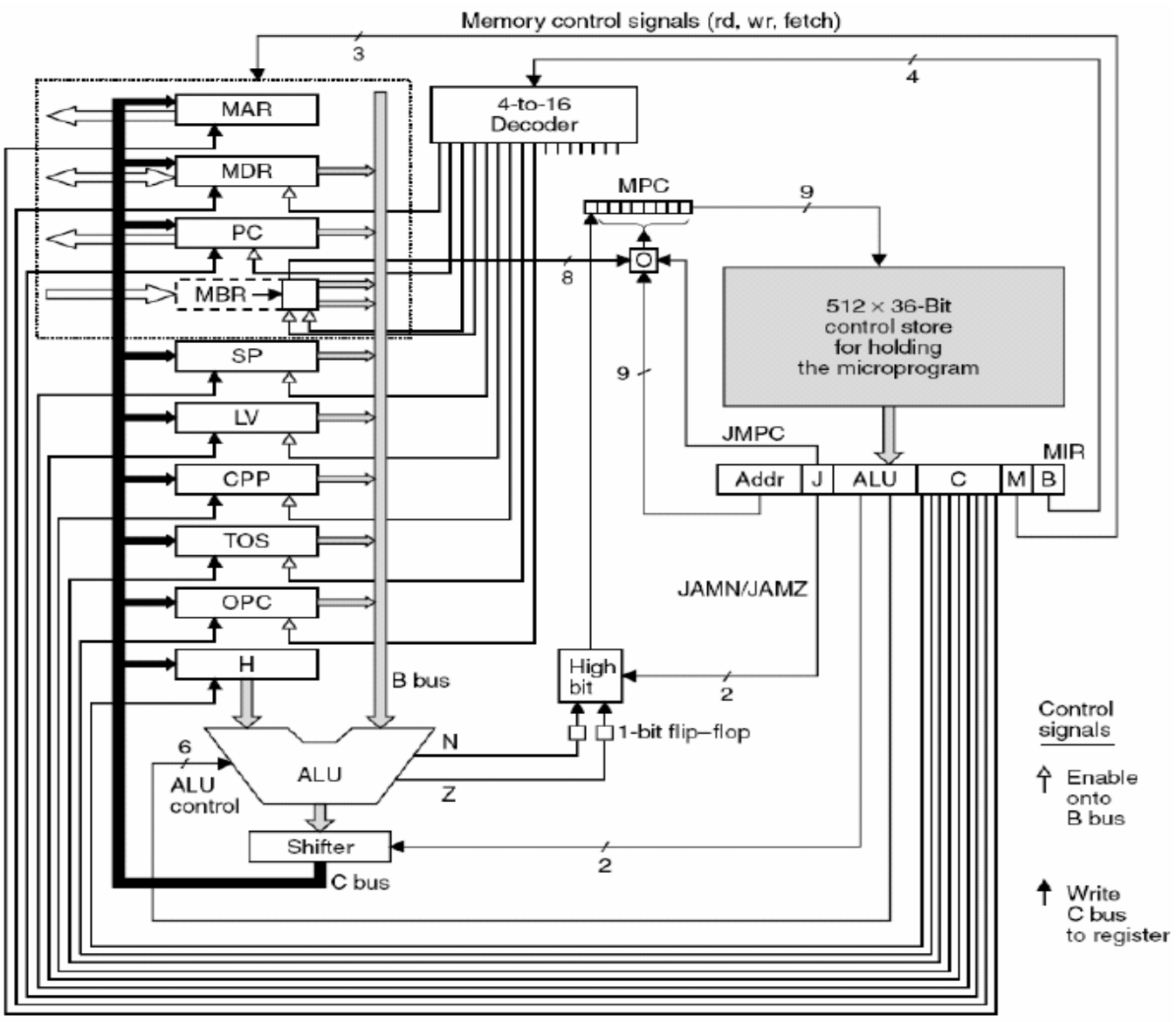
b)
Figure 9 shows the original IJVM. Figure 13 shows a version of IJVM with an extra bus (A bus) and a instruction fetch unit as to improve performance.

i) How does the extra bus (A bus) influence on the execution of instructions? Explain short.
ii) Is it necessary to change the micro programs for the instructions to exploit the new bus? Explain short

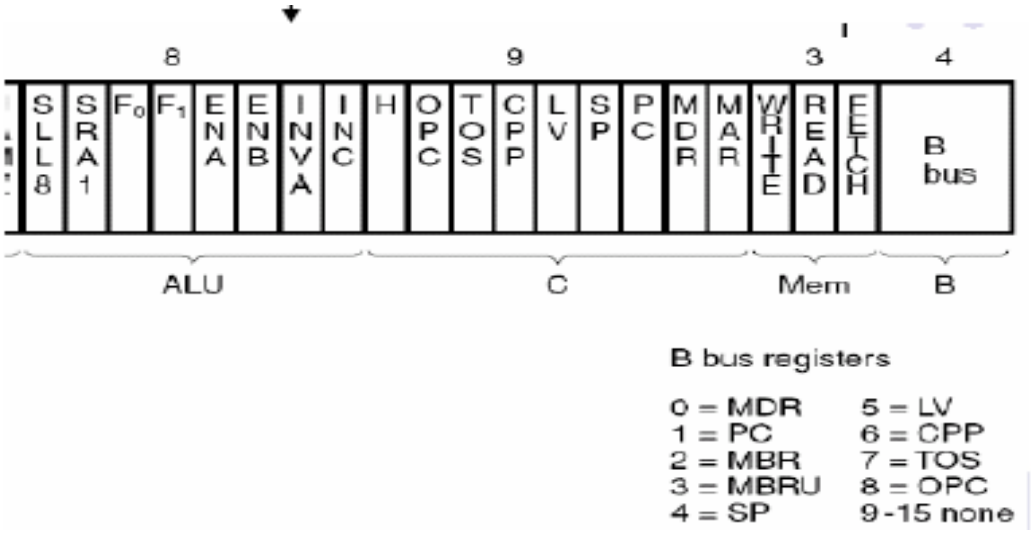
c)
What development is sketched out in Moore's law?

d)
What parameter was decrease to reduce energy consumption when moving from single core processors with deep pipelines to CMP based processors?

# Enclosed IJVM docs




Figur 9 IJVM

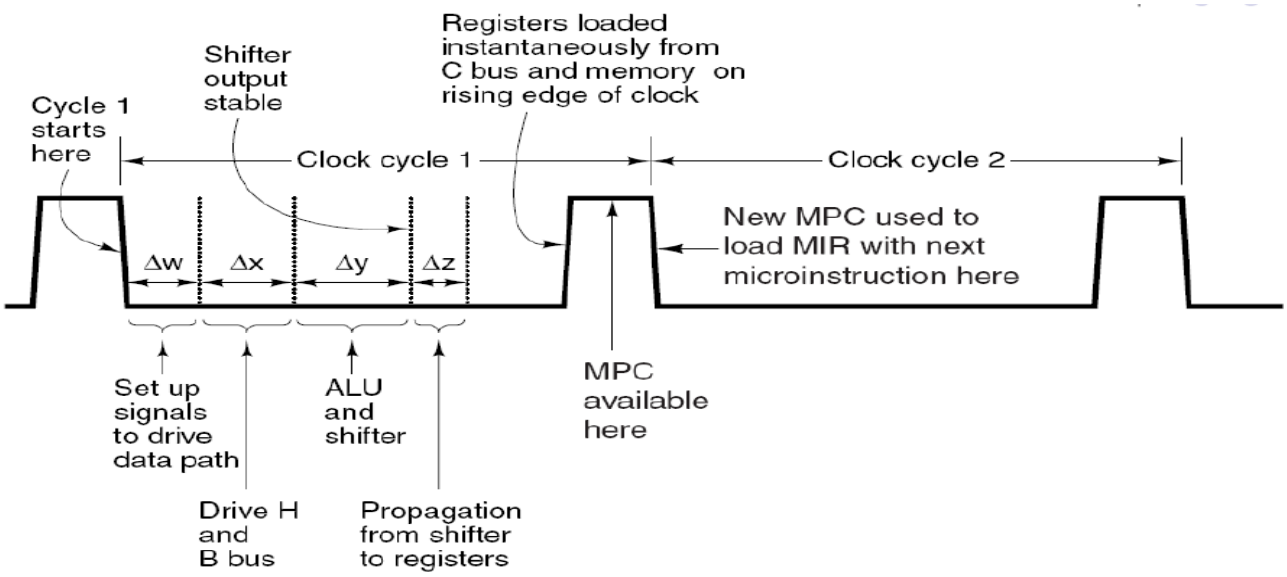



Figur 10 Microinstruction format.

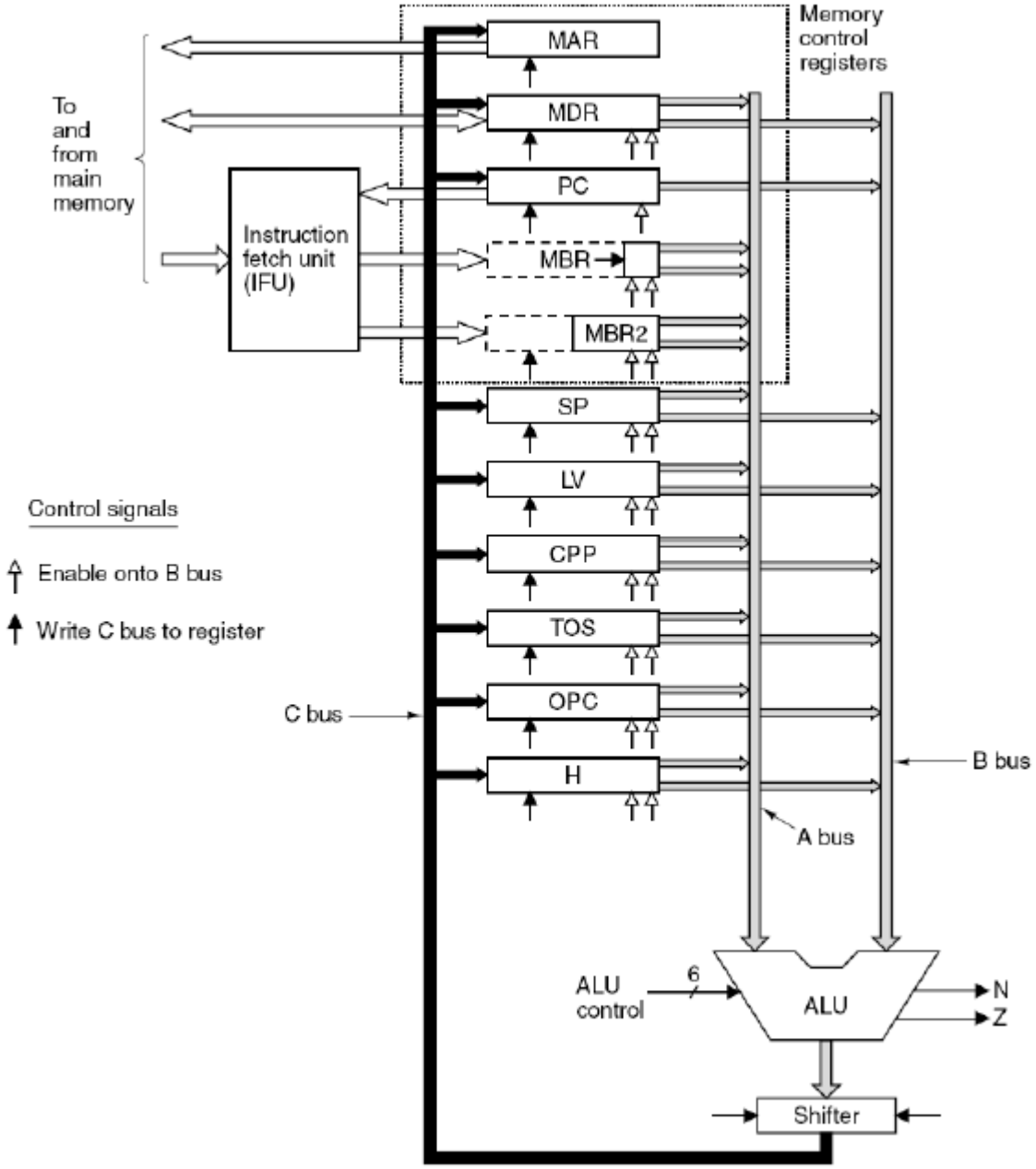
| $F_0$ | $F_1$ | ENA | ENB | INVA | INC | Function |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | A |
| 0 | 1 | 0 | 1 | 0 | 0 | B |
| 0 | 1 | 1 | 0 | 1 | 0 | $\overline{A}$ |
| 1 | 0 | 1 | 1 | 0 | 0 | $\overline{B}$ |
| 1 | 1 | 1 | 1 | 0 | 0 | A + B |
| 1 | 1 | 1 | 1 | 0 | 1 | A + B + 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | A + 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | B + 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | B – A |
| 1 | 1 | 0 | 1 | 1 | 0 | B – 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | –A |
| 0 | 0 | 1 | 1 | 0 | 0 | A AND B |
| 0 | 1 | 1 | 1 | 0 | 0 | A OR B |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | –1 |

| SLR1 | SLL8 | Function |
|---|---|---|
| 0 | 0 | No shift |
| 0 | 1 | Shift 8 bit left |
| 1 | 0 | Shift 1 bit right |

Figur 11 ALU functions.




Figur 12 iming diagram.




Figur 13 MIC 2.