

Lecture 0

# BSCS1004 Computer Organization I (TKT10005)

Teemu Kerola



Course Introduction  
Learning Goals  
Course Components

# Computer Organization (I)

- Course contents and learning goals
- Course components
  - Lectures
  - Homework exercises, practice sessions
  - Ttk -91 programming workshop
  - Mini exams
- Ttk-91 example computer system
  - Titokone simulator
  - TitoTrainer practice environment
- Read all www-info carefully

titopaja

<https://courses.helsinki.fi/fi/aytk100052/132554318>

<https://www.cs.helsinki.fi/group/nodes/kurssit/tito/2020k/schedule.html>

# Course View Point: Execute One Program

- What is the representation of a program in system?
- How does the hardware (HW) execute the program?
- Which HW components participate and how?
- Goal: general idea on how a computer system operates and its basic structure

Discuss

# Computer System

Java program

Programming environment  
(E.g., Net Beans)

Operating System  
(E.g., Linux/Windows)

CO-I



Turing

Von Neumann

Tito  
(CO-I)

CPU

Display

Keyboard

Memory

Hard disk

Boole

ports

clock pulse

electricity

power

micro chip

# Computer System as Numbers

- Examples on processors:
  - AMD Athlon II X4 645, 3.1 GHz
  - Intel Core i5-520U, 2.4 GHz
  - ARM Cortex A-8, 600 MHz – 1 GHz

- Main memory

- 4 – 500 GB (giga byte)
- KB, MB, GB, TB

The numbers for current technology are not important!

- What GHz means?

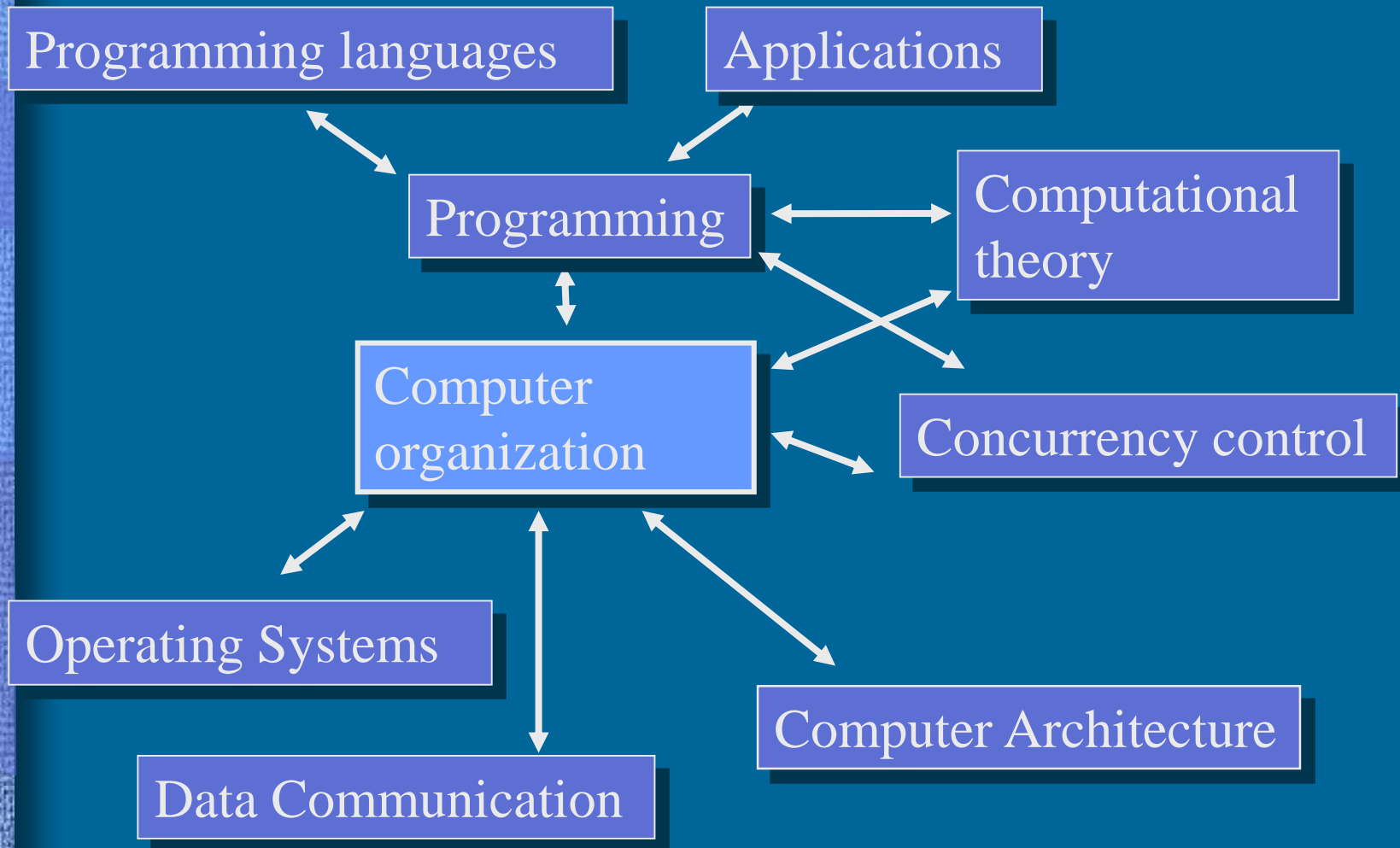
- How many bits are needed for memory address?

# SI-system Prefixes

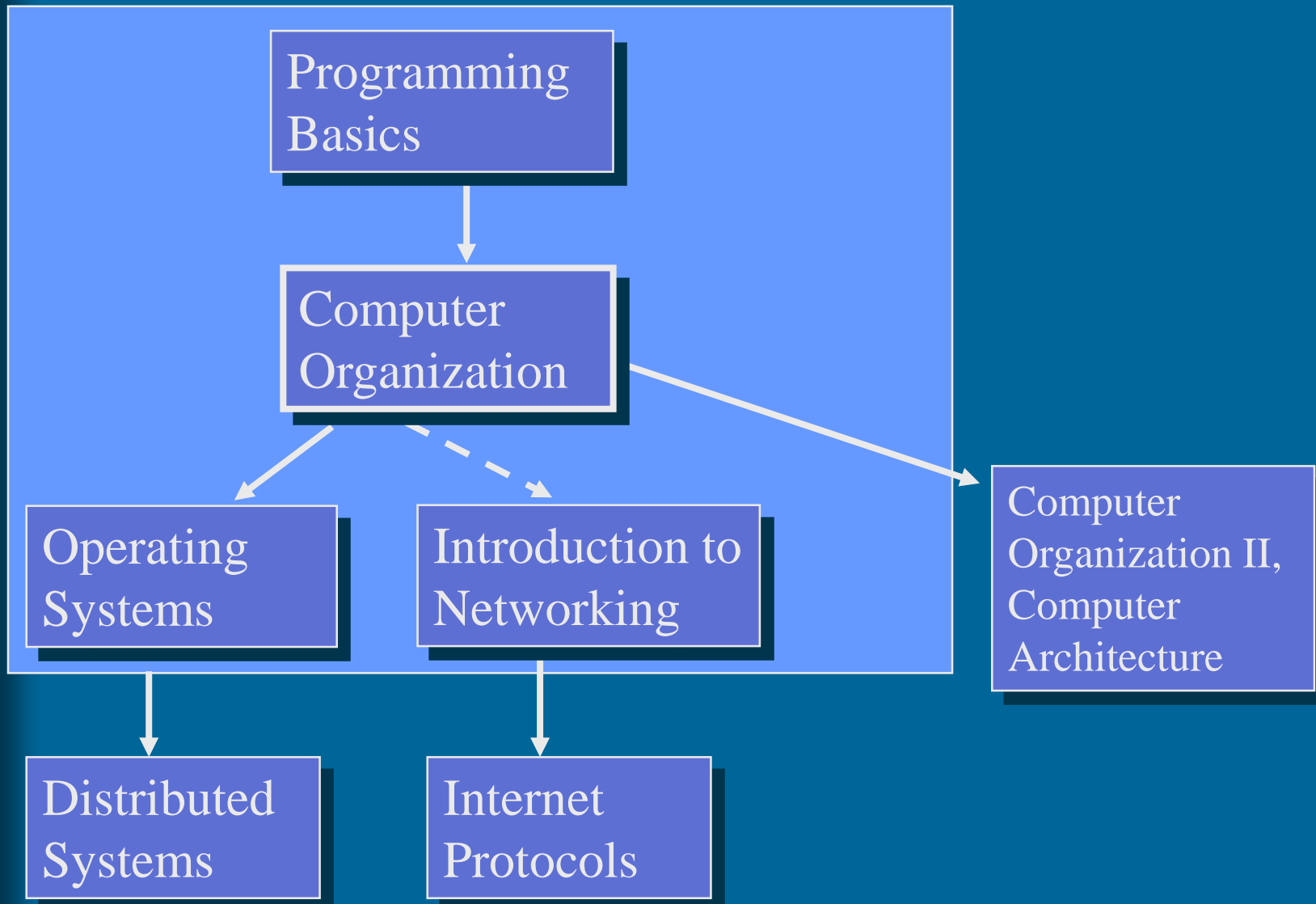
kerrannaisyksiköt

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
$10^{-3}$	0.001	milli	$10^3$	1,000	Kilo
$10^{-6}$	0.000001	micro	$10^6$	1,000,000	Mega
$10^{-9}$	0.000000001	nano	$10^9$	1,000,000,000	Giga
$10^{-12}$	0.0000000000001	pico	$10^{12}$	1,000,000,000,000	Tera
$10^{-15}$	0.0000000000000001	femto	$10^{15}$	1,000,000,000,000,000	Peta
$10^{-18}$	0.0000000000000000001	atto	$10^{18}$	1,000,000,000,000,000,000	Exa
$10^{-21}$	0.00000000000000000000001	zepto	$10^{21}$	1,000,000,000,000,000,000,000	Zetta
$10^{-24}$	0.0000000000000000000000001	yocto	$10^{24}$	1,000,000,000,000,000,000,000,000	Yotta

# Dependencies Between Topics



# Dependencies between Courses





# Computer Organization I (5 cr)

TKT10005

- Lecture course in Finnish
  - Lectures (in Finnish)
  - MOOC-materials (in Finnish, 2 cr + 3 cr)
  - Titokone/TitoTrainer in workshop and independ. (Finnish/English)
  - Homework exercises, practice sessions (also in English)
  - Mini exams (4, also in English)
- Self-study
  - Same topics is in most previous lecture course
    - MOOC-materials (in Finnish)
    - Text book (in English)
  - Titokone/TitoTrainer independently
  - Separate exam
- Lecture course in English
  - Lectures in English
  - Text book (in English)
  - Titokone/TitoTrainer in workshop and independ. (English/Finnish)
  - Homework exercises, practice sessions (partly also in Finnish)
  - Mini exams (4, also in Finnish)

”ordinary course  
in Finnish”

”self-study”

BSCS1004

this course

”ordinary course  
in English”

Same  
course,  
either  
one will  
do for  
degree  
require-  
ments

# Lectures

- On given topic (see daily schedule)
  - Study in advance from text book
- Short small group chats concerning some of the lecture topics
- Ask questions

# Homework Exercises and Practice Sessions (12/60 p)

- Ordinary homework given in daily schedule page
- Learning happens when solving problems and discussing solutions with your peers
  - Study first and only then do the homework (alone or with friend)
  - Work on the homework exercise yourself before discussing it
  - Just reading complete solution is a waste of good learning opportunity
  - Giving complete solution to other student will hamper his/her learning
- Counts toward your course grade
  - Homework points (hwp) from exercises you (mostly) solved
  - Only for those present in practice meetings
  - Hwp's are mapped linearly to grade points (some 83% → 12 p)

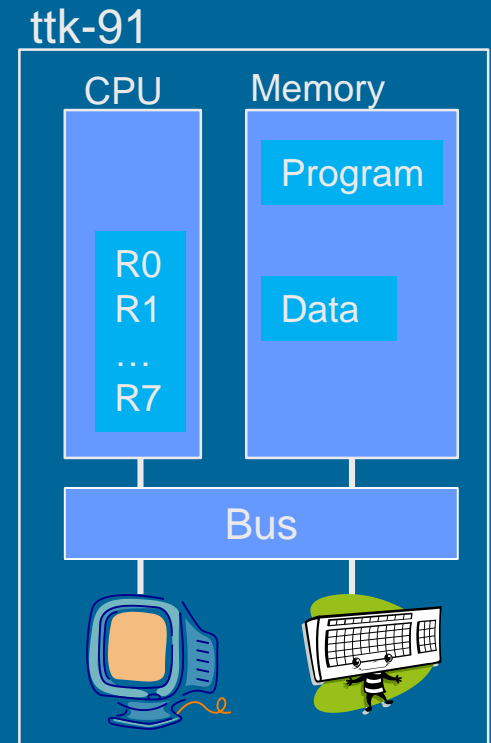
# Practice Sessions



- Students are split to small groups (“tables”)
  - Hopefully each group has for each problem someone, who has solved it
- All problems are discussed through in the group
  - No need to go through all good solutions from each student
  - Can assume, that everyone is familiar with problems and at least tried to solve them
  - Everyone should get understanding on (one) good solution
  - Instructor helps when needed
- Model solutions are handed out
  - They are discussed in each group
  - Some or all problems are discussed together
- Everyone is present until session ends

# Ttk-91 Example Computer System

- Auvo Häkkinen, 1991
  - Tietokoneen toiminta –kurssi 1991  
(Computer Organization Course 1991)
- Simple computer architecture
  - Abstraction level just right for this course
  - Processor, system, operating system
- Simple (symbolic) assembly language
  - Easy to learn, not too many instructions
  - Just right for this course
- Learning goals
  - Understand, what type of code processor uses
  - Understand, how the computer system executes code



# Titokone

- Java program, that simulates ttk-91 and its operating system
  - ttk-91 specification, Auvo Häkkinen, 1991
    - Orig. Ttk-91 simulator was written in Pascal, not used any more
  - Operates the same way as hardware implemented ttk-91 system and its operating system
  - Software Dev Project “Koski”, Spring 2004

hello.k91



hello.b91

## Titokone

- Ttk-91 symbolic assembly language compiler
- Ttk-91 emulator (simulator), which can execute compiled (type .b91) ttk-91 symbolic assembly programs
- Software development environment built into the emulator
- Animator visualizes emulated instruction execution in ttk-91
- Graphical user interface

```
LOAD R3, X  
→ 0x39845890
```

<http://www.cs.helsinki.fi/group/nodes/kurssit/tito/esimerkit/>

<http://www.cs.helsinki.fi/group/titokone/distr/titokone-1.203.jar>

# TitoTrainer

<http://titotrainer2.users.cs.helsinki.fi>

- Current version: TitoTrainer2
- Software built on top of Titokone
- Implement ttk-91 programs or sections of them
  - Same programs can be run in Titokone
- Program correctness is checked automatically
- Ttk-91 programming workshop in course week 4/5?
- Homework exercisers in course week 4 and 5
  - hw3 and hw4
- Mini exam 3

# Exams (48/60 p)

- All course exams are mini exams (4 exams, each 12 p)
  - Given during lectures (45 min)
  - Wednesdays 25.3.2020, 1.4.2020, 22.4.2020, 29.4.2020
    - at 14:15-15:00 (must stay until the end of exam)
    - me1, me2, me3, me4
  - Topics: preceding lectures and homework (see schedule)
- Replacement exams
  - You can redo all (1-4) mini exams with June final exam
  - Maximum of points for each mini exam counts
  - No other possibilities for replacement exams



# Grading

Homework exercises, practice sessions	12 p
Mini exams (replacement mini exams) <i>minimum level to pass 24/48 p</i>	48 p
<b>Total</b> <i>minimum level to pass 30/60 p</i>	60 p

# Learning Material

- Text books

<https://helka.finna.fi/>

- Stallings: Computer Org. and Architecture, 10th ed., 2016 (8<sup>th</sup>, 9<sup>th</sup>, and 11<sup>th</sup> ed. ok)
  - May be available in E-book versions (9<sup>th</sup> ed?)
    - Limited nr of licenses?
- Tanenbaum: Structured Computer Organization, 6th Ed, 2013 (5th ed. ok)

- Lectures

- Titokone and TitoTrainer environments

- Homework exercises and practice sessions

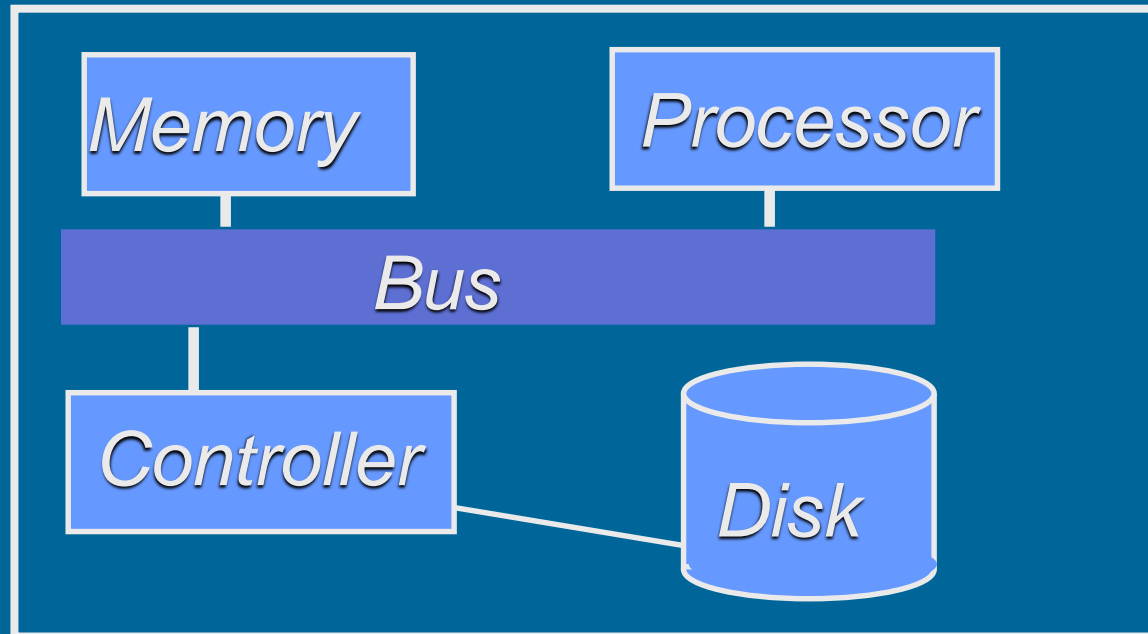
# WWW Information

- **Course page** <https://courses.helsinki.fi/en/bscs1004/130382925>
  - Specific info for this lecture course (Spring 2020)
- **Course info** <https://courses.helsinki.fi/en/bscs1004>
  - General info for this course
- **Course Schedule** <https://www.cs.helsinki.fi/group/nodes/kurssit/tito/2020k/schedule.html>
  - Precise daily schedule
    - Lectures, practice sessions, cs1-workshops, mini exam
  - Links
    - Lecture notes (slides), exercises, exams, solutions
    - Course page, ttk-91, Titokone, TitoTrainer

# Course Contents

- Structure of computer system
- CPU and bus
- Data representation
- Protecting data integrity
  
- Ttk-91 computer system and its simulator
- Assembly language programming
- Subroutines in general and in assembly language programming
  
- Operating system and process
- External memory, I/O implementation
- Compilation, linking, loading
- Interpretation, emulation, Java-program execution

# Co-I vs. CO-II



**CO-I:** What happens in system?  
What does OS do?

**CO-II:** How is CPU and memory implemented?  
How will clock tick make it all happen?

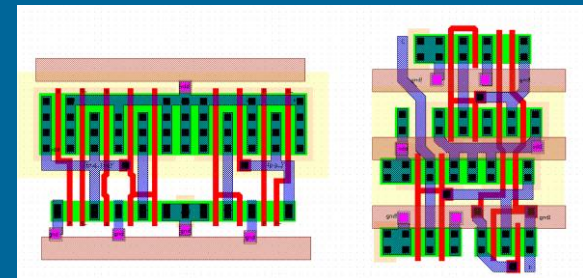
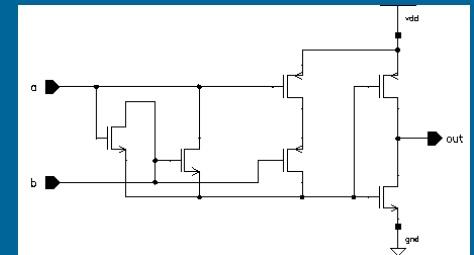
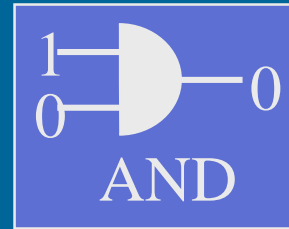
# Processor Implementation Hierarchy

- Machine lang architecture
  - ADD R1, R2
- Modules
  - Adder, register, ALU
- Logical gates
  - AND, OR, NOR
- Circuit design
  - Power consumption, timing, wire layout
- Implementation technology
  - Vacuum tube, transistor, micro chip

Rest of text book ...

CO-II

CO-I



# Motto

- “It is not good exercise, if you do not sweat”  
– However, this is not a marathon!



- Use some 12 h / week  
+ preparations for exams + exams

- Or:  $5 \text{ yrs} / 300 \text{ cr} = 1 \text{ yrs} / 60 \text{ cr} = 1600 \text{ h} / 60 \text{ cr}$   
 $= 26.67 \text{ h} / 1 \text{ cr} = \underline{133 \text{ h}} / 5 \text{ cr}$

