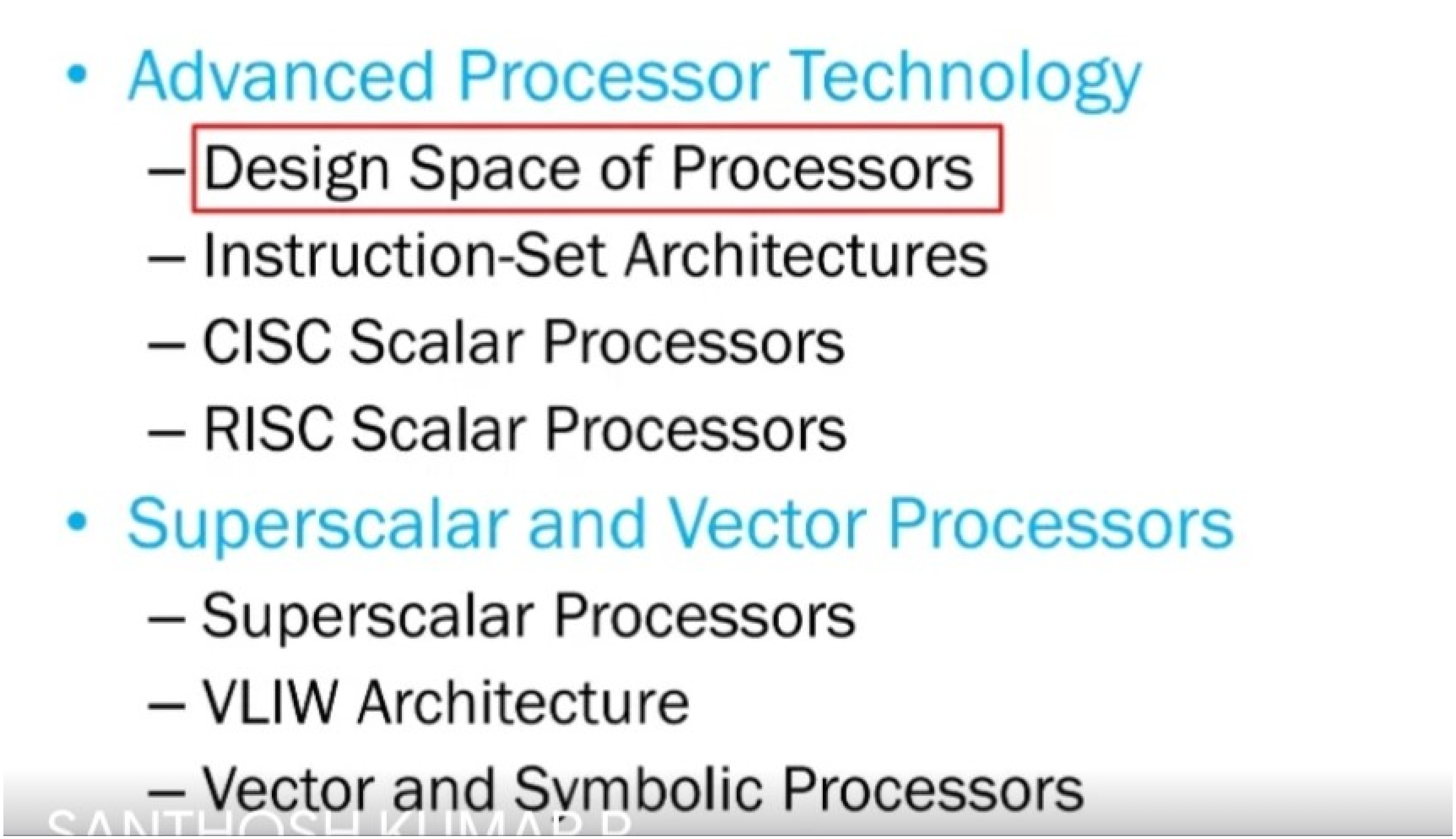
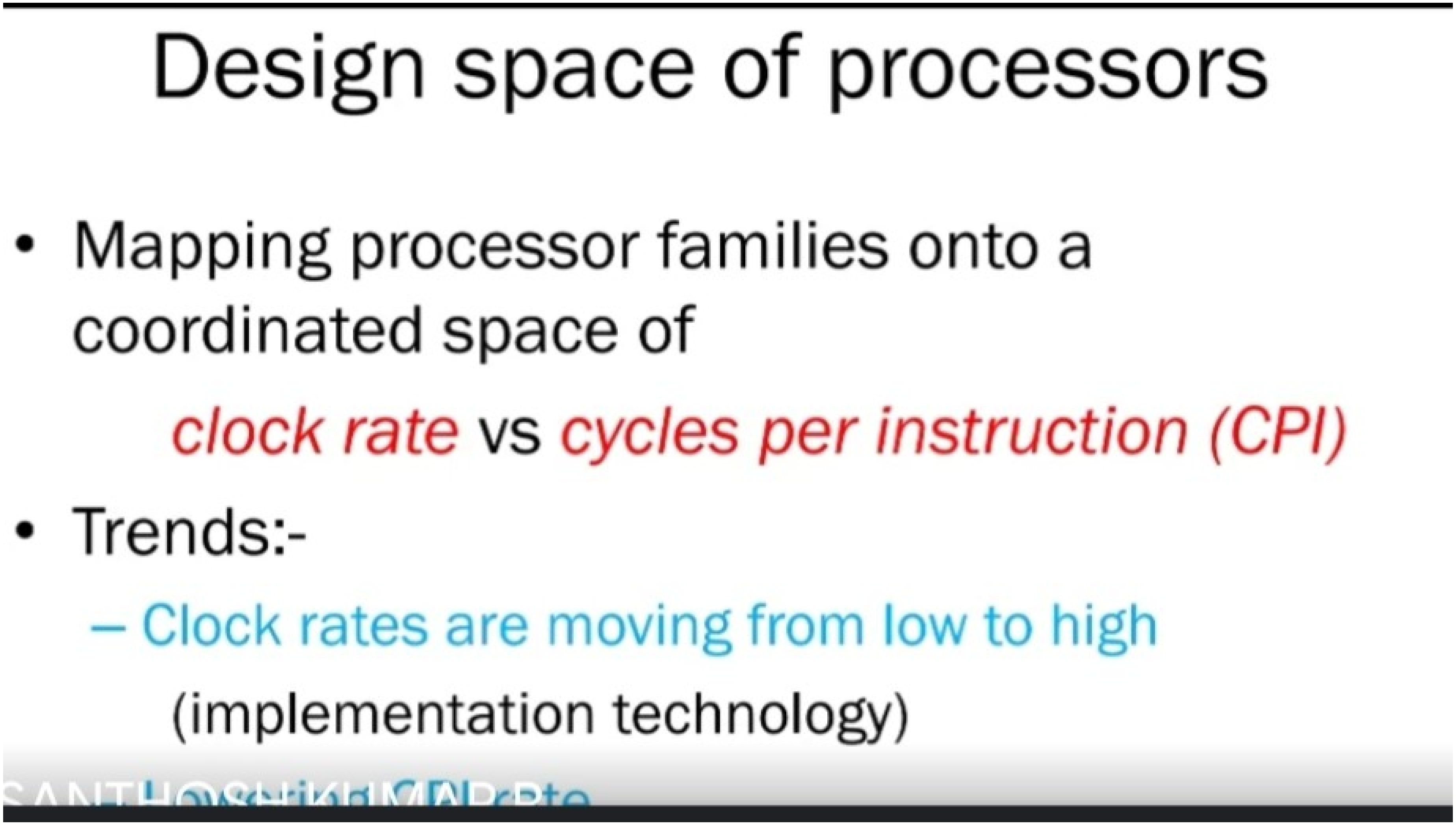


- Advanced Processor Technology
  - Design Space of Processors
  - Instruction-Set Architectures
  - CISC Scalar Processors
  - RISC Scalar Processors
- Superscalar and Vector Processors
  - Superscalar Processors
  - VLIW Architecture
  - Vector and Symbolic Processors



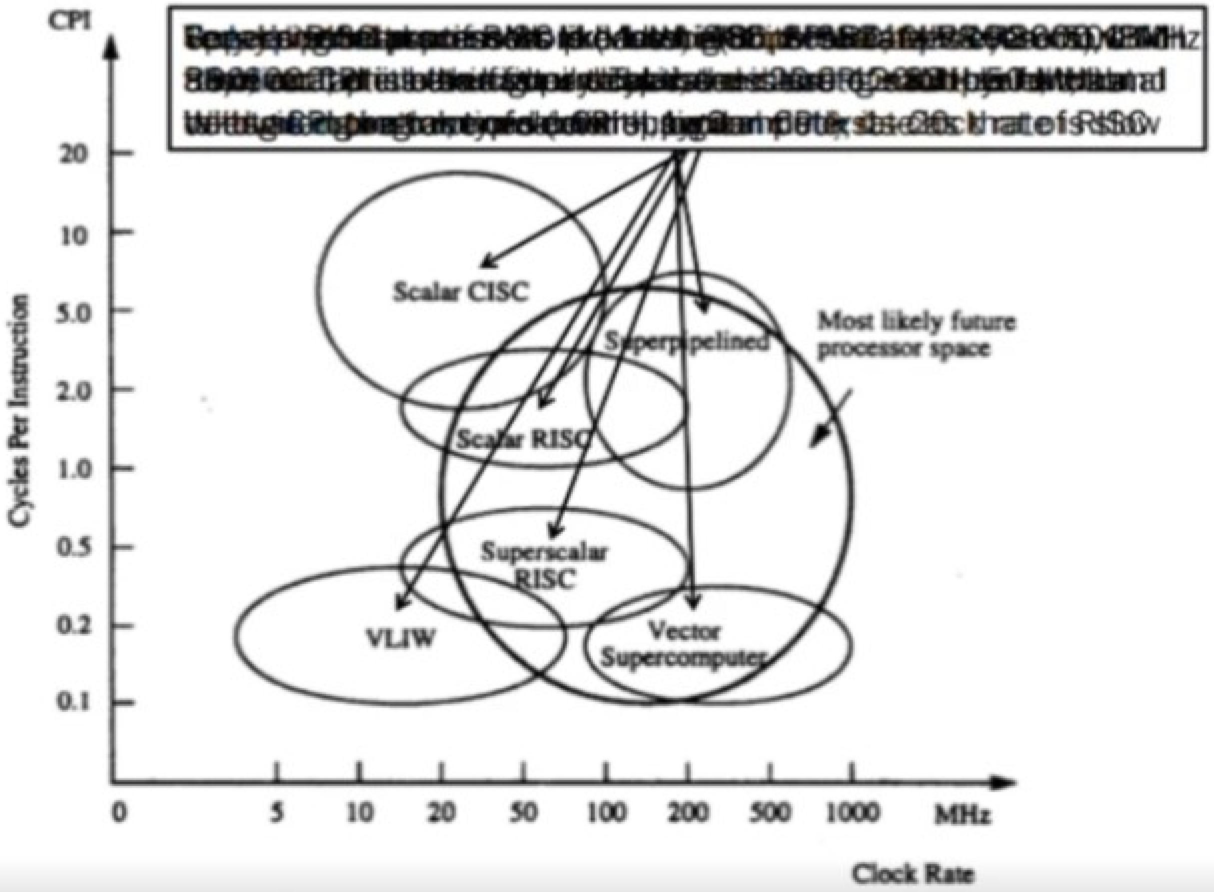
Design space of processors

- Mapping processor families onto a coordinated space of

  clock rate vs *cycles per instruction (CPI)*

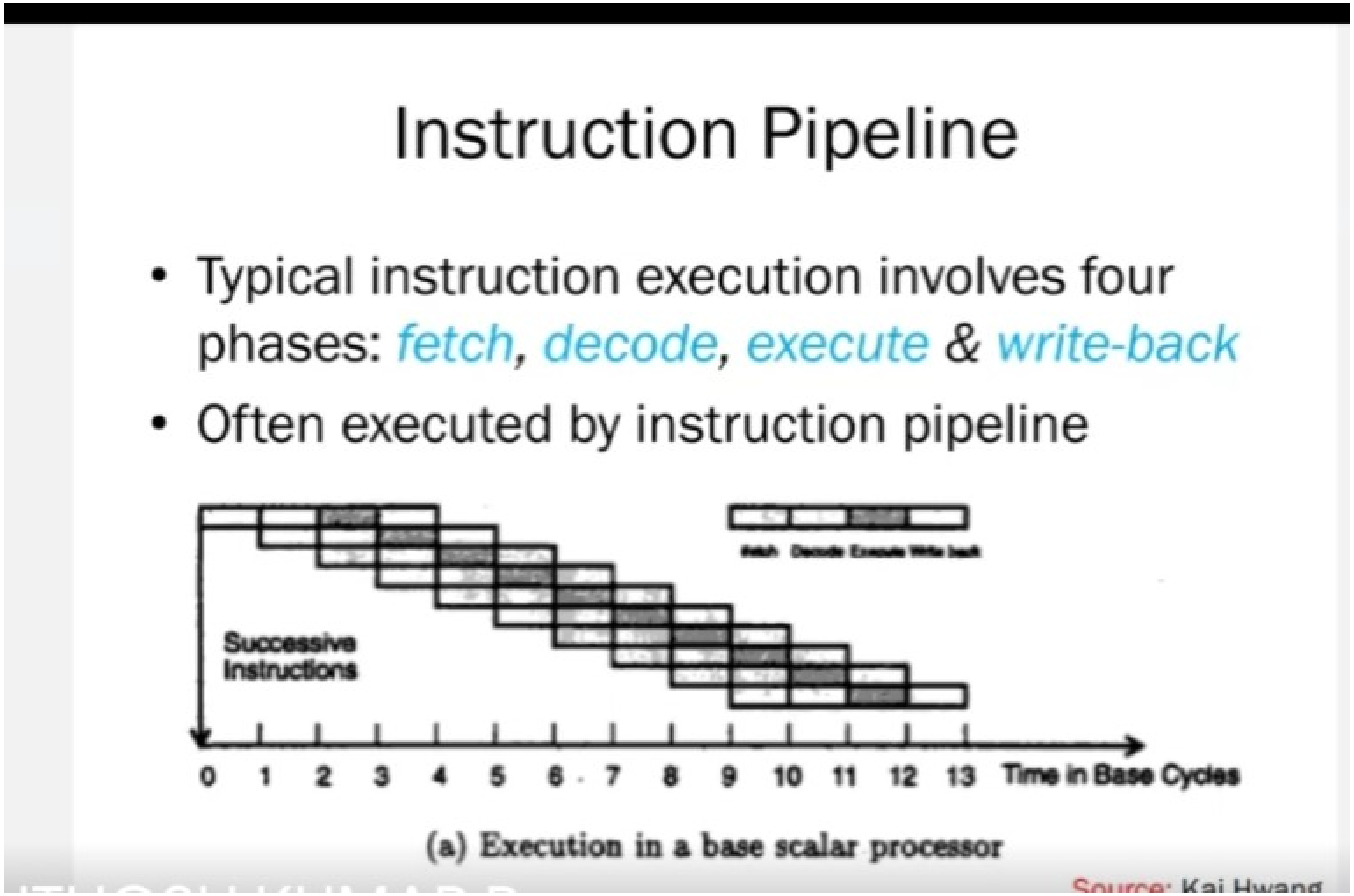
- Trends:-
  - Clock rates are moving from low to high

    (implementation technology)



[illegible]

THOSH KUMAR P

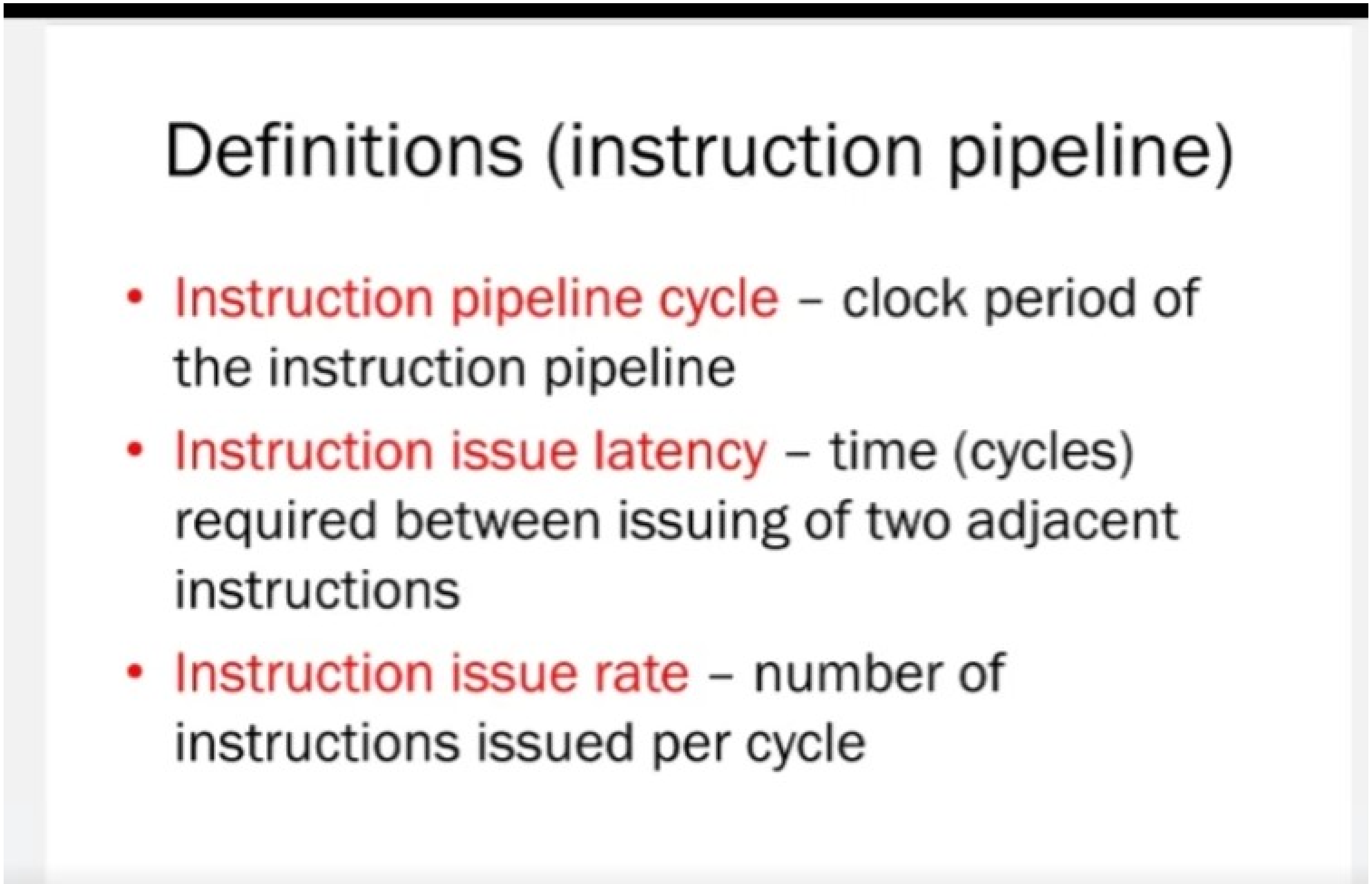


Instruction Pipeline

- Typical instruction execution involves four phases: *fetch*, *decode*, *execute* & *write-back*
- Often executed by instruction pipeline

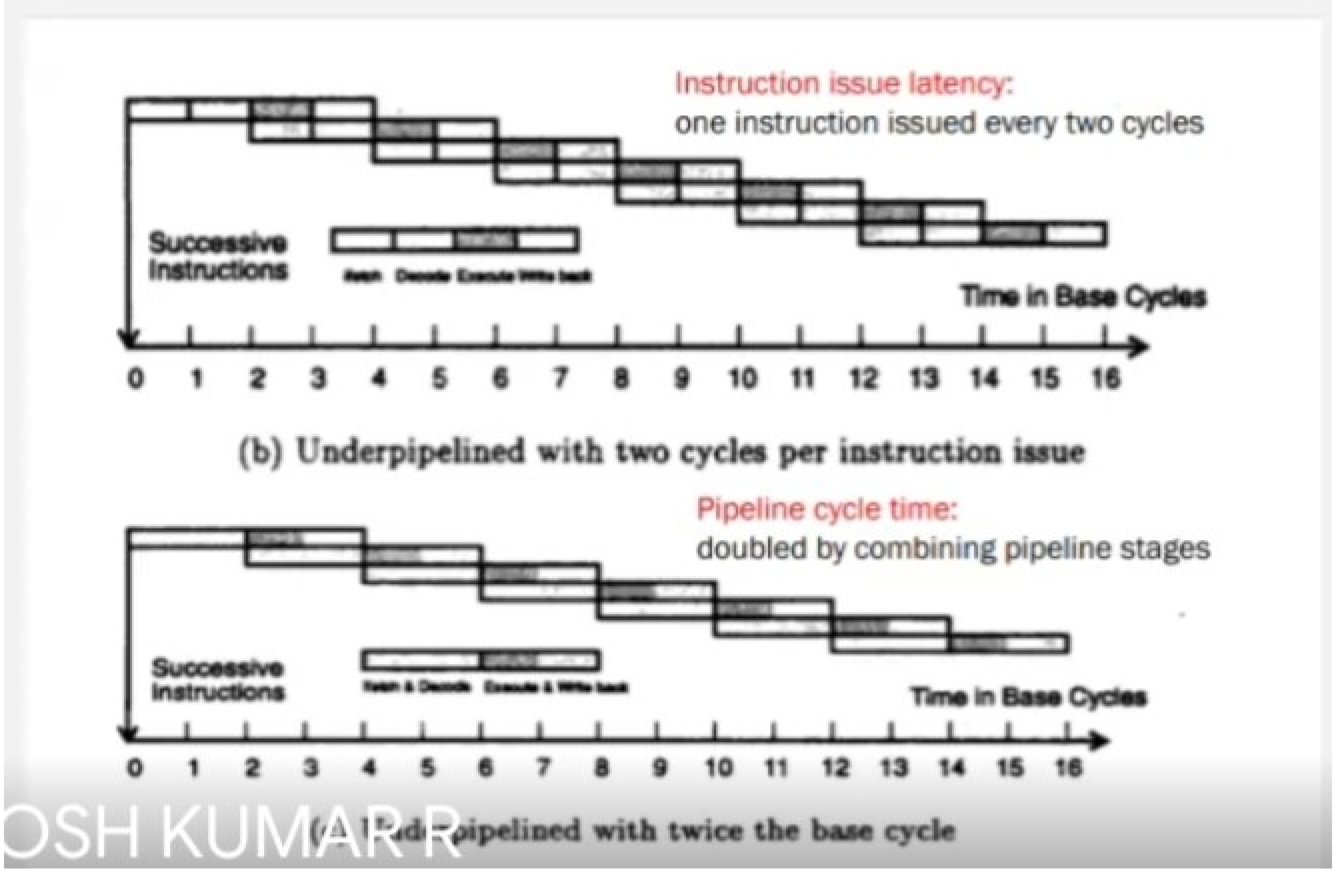
(a) Execution in a base scalar processor

Source: Kai Hwang



Definitions (instruction pipeline)

- Instruction pipeline cycle – clock period of the instruction pipeline
- Instruction issue latency – time (cycles) required between issuing of two adjacent instructions
- Instruction issue rate – number of instructions issued per cycle

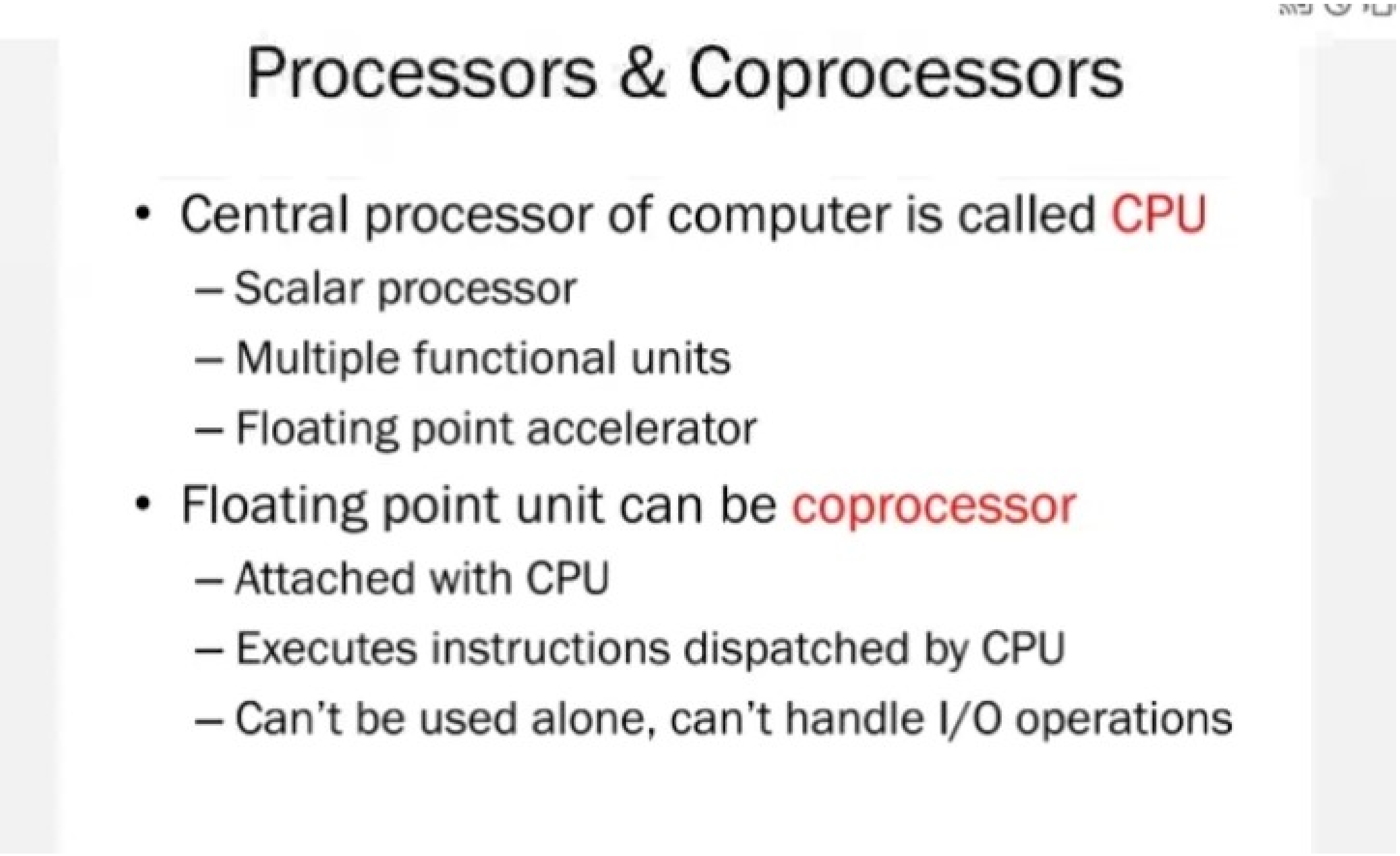


Instruction issue latency:
one instruction issued every two cycles

(b) Underpipelined with two cycles per instruction issue

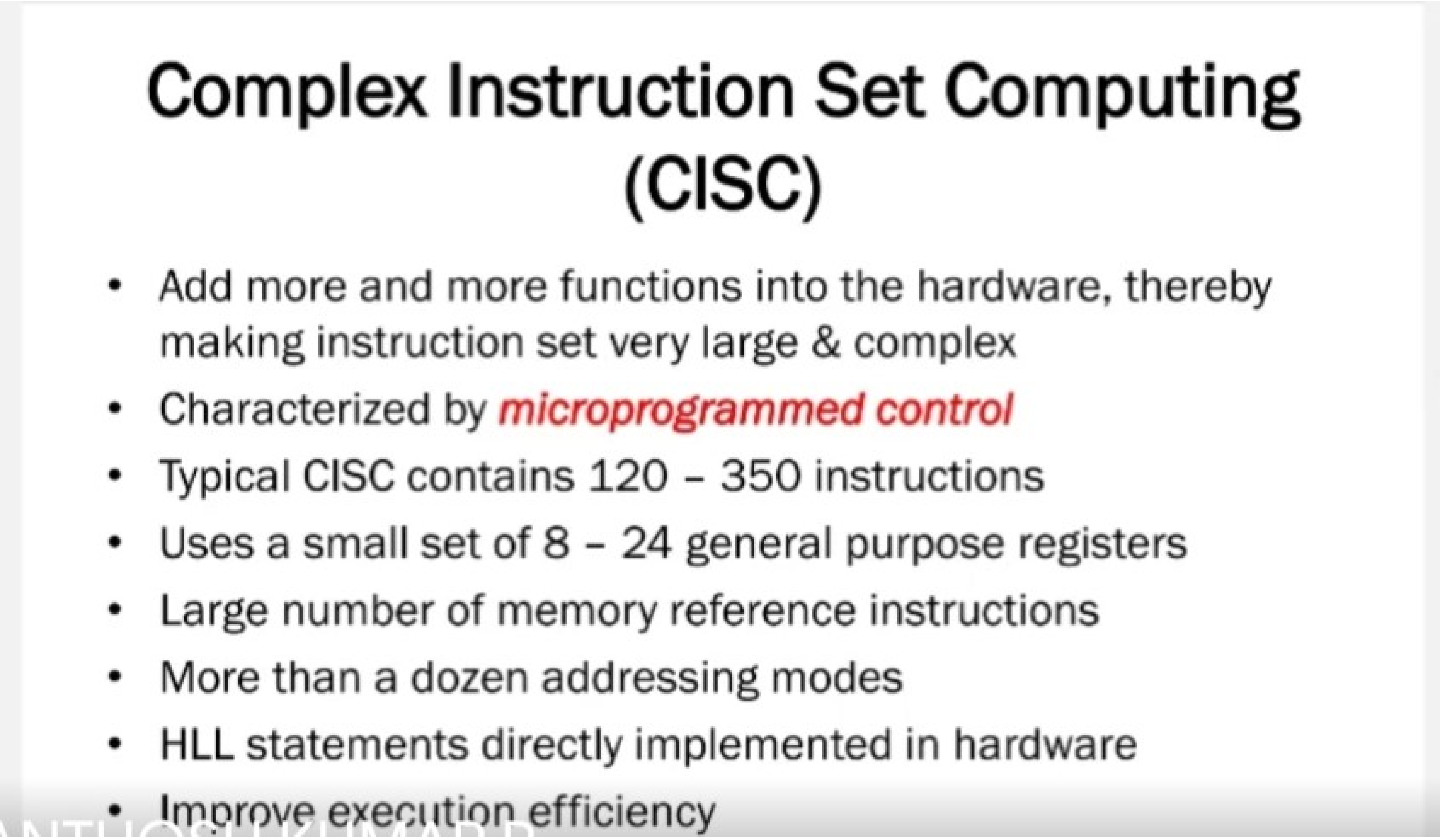
Pipeline cycle time:
doubled by combining pipeline stages

(c) Underpipelined with twice the base cycle



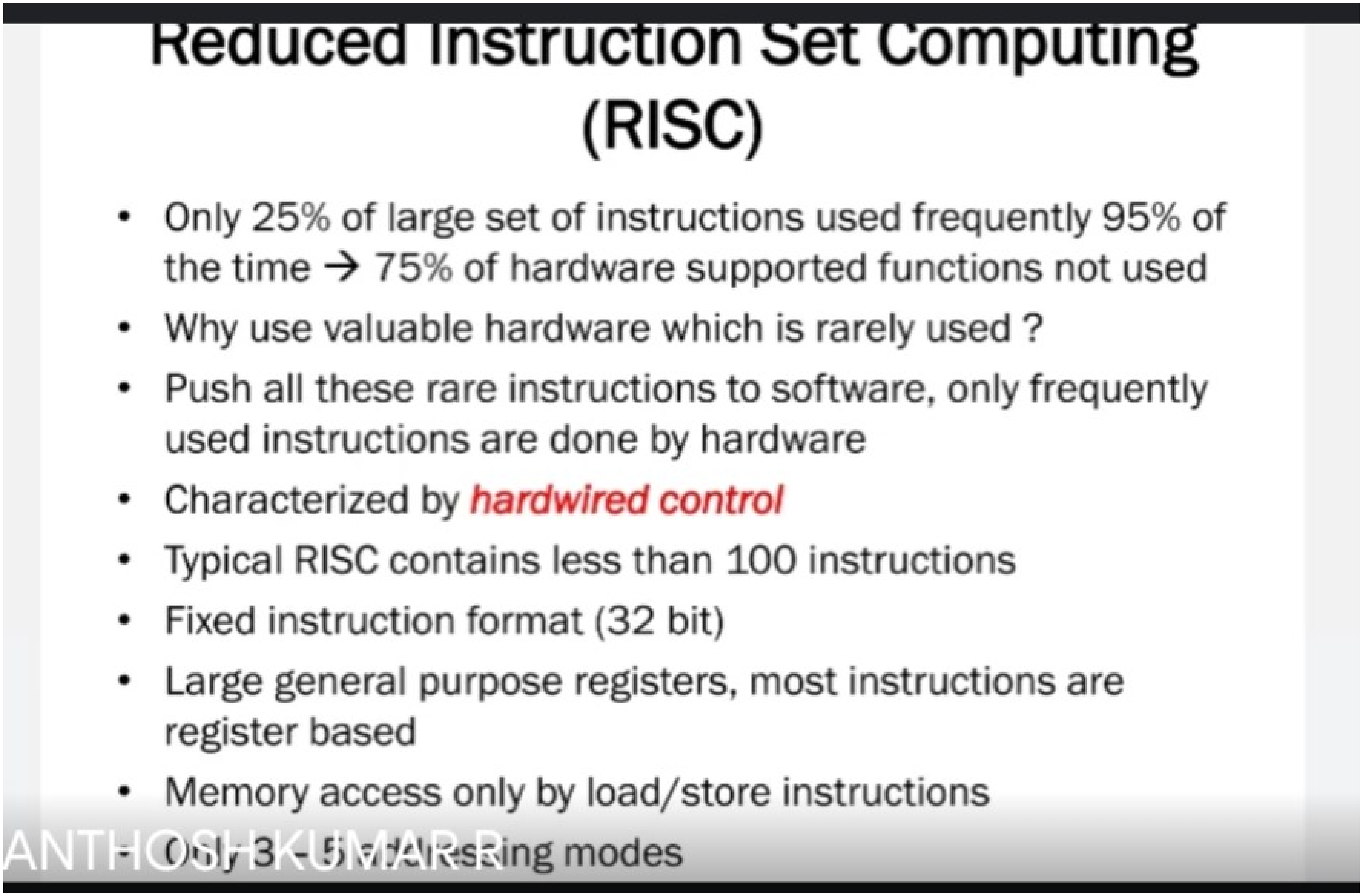
Processors & Coprocessors

- Central processor of computer is called CPU
  - Scalar processor
  - Multiple functional units
  - Floating point accelerator
- Floating point unit can be coprocessor
  - Attached with CPU
  - Executes instructions dispatched by CPU
  - Can’t be used alone, can’t handle I/O operations



Complex Instruction Set Computing (CISC)

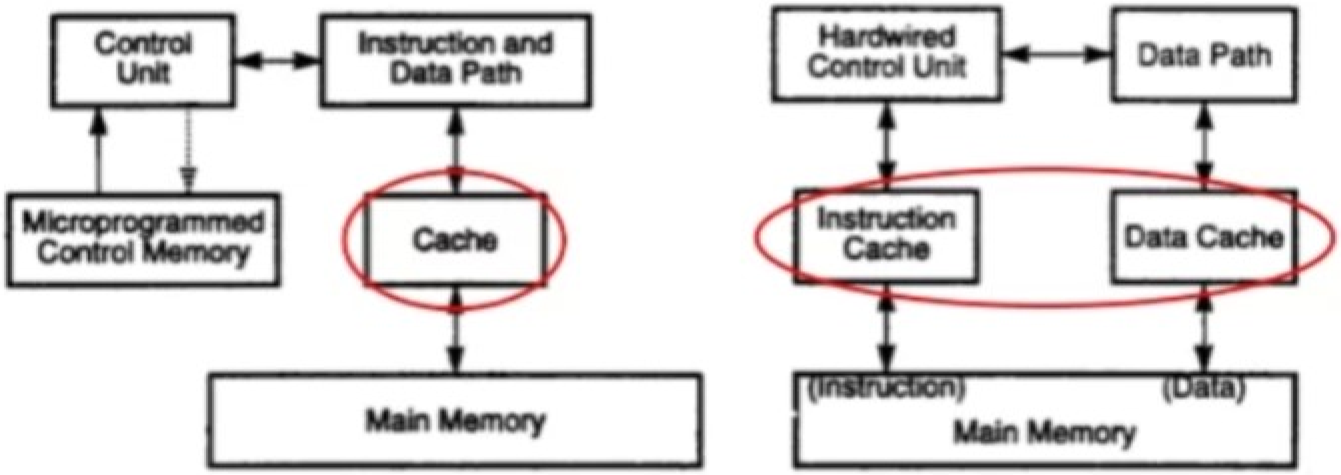
- Add more and more functions into the hardware, thereby making instruction set very large & complex
- Characterized by ***microprogrammed control***
- Typical CISC contains 120 – 350 instructions
- Uses a small set of 8 – 24 general purpose registers
- Large number of memory reference instructions
- More than a dozen addressing modes
- HLL statements directly implemented in hardware
- Improve execution efficiency



Reduced Instruction Set Computing (RISC)

- Only 25% of large set of instructions used frequently 95% of the time → 75% of hardware supported functions not used
- Why use valuable hardware which is rarely used ?
- Push all these rare instructions to software, only frequently used instructions are done by hardware
- Characterized by ***hardwired control***
- Typical RISC contains less than 100 instructions
- Fixed instruction format (32 bit)
- Large general purpose registers, most instructions are register based
- Memory access only by load/store instructions
- Only 3 - 5 addressing modes

CISC vs RISC Architectures



(a) The CISC architecture with microprogrammed control and unified cache

(b) The RISC architecture with hardwired control and split instruction cache and data cache.

Figure 4.4 Distinctions between typical RISC and typical CISC processor architectures. (Courtesy of Gordon Bell, 1989)

Source: Kai Hwang

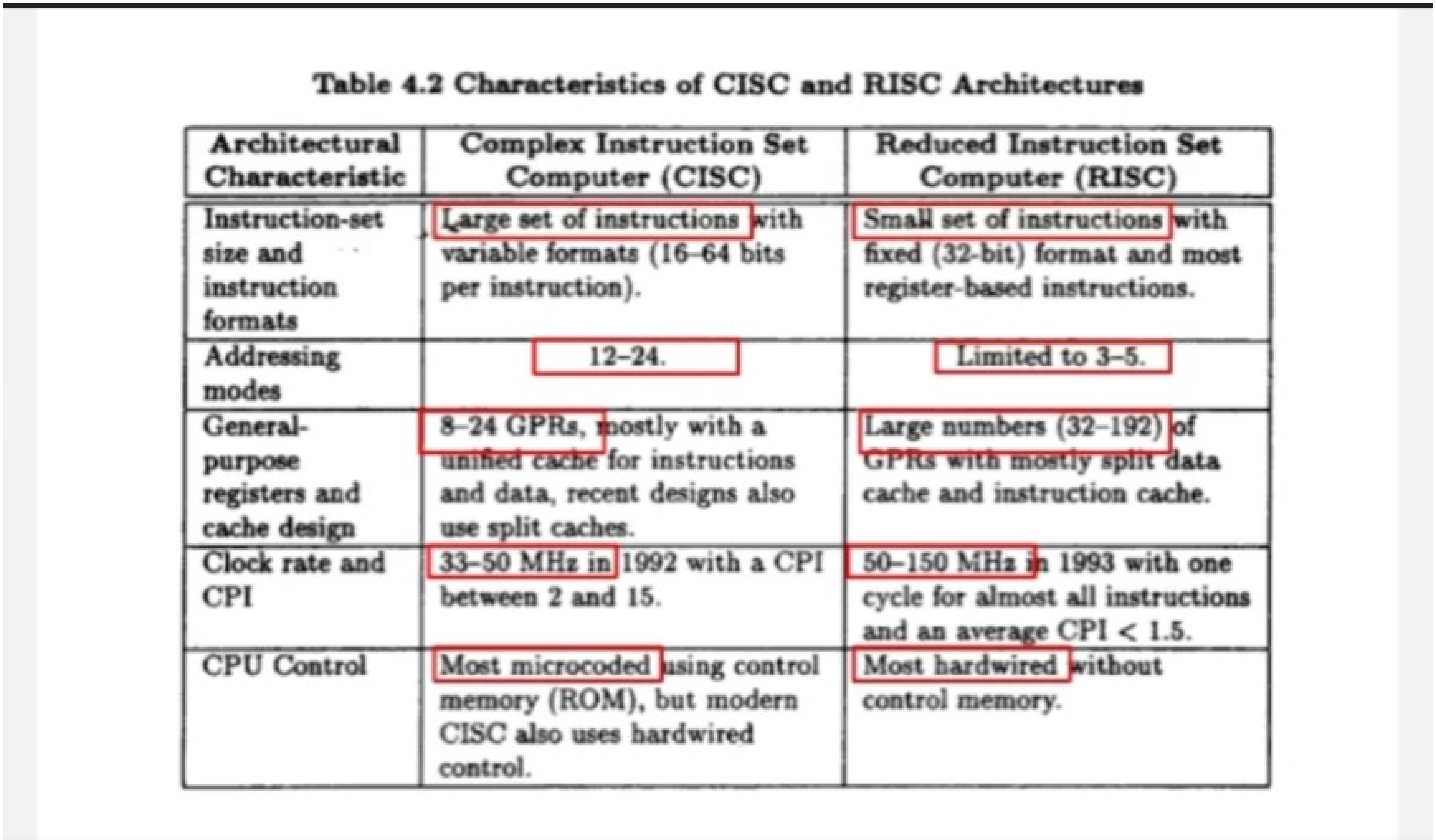


Table 4.2 Characteristics of CISC and RISC Architectures

Architectural Characteristic	Complex Instruction Set Computer (CISC)	Reduced Instruction Set Computer (RISC)
Instruction-set size and instruction formats	Large set of instructions with variable formats (16–64 bits per instruction).	Small set of instructions with fixed (32-bit) format and most register-based instructions.
Addressing modes	12–24.	Limited to 3–5.
General-purpose registers and cache design	8–24 GPRs, mostly with a unified cache for instructions and data, recent designs also use split caches.	Large numbers (32–192) of GPRs with mostly split data cache and instruction cache.
Clock rate and CPI	33–50 MHz in 1992 with a CPI between 2 and 15.	50–150 MHz in 1993 with one cycle for almost all instructions and an average CPI < 1.5.
CPU Control	Most microcoded using control memory (ROM), but modern CISC also uses hardwired control.	Most hardwired without control memory.