CPE224 Computer Architecture

Intel® Core™ i7-8700K

สมาชิกกลุ่ม

1.นาย ปวริศ ร้านชิตวงศ์ 61070501034

2.นาย ปิยวิทย์ ถิระสินสมบูรณ์ 61070501036

3.พีรภัทร เขมะชิต 61070501039

นักศึกษาชั้นปีที่ 2 คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

นำเสนอ

อาจารย์ ราชวิชช์ สโรชวิกสิต

รายงานนี้เป็นส่วนหนึ่งของวิชา CPE 224 COMPUTER ARCHITECTURES

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา2562

Intel® Core™ i7-8700K

Essentials information

Product Collection	8th Generation Intel® Core™ i7 Processors
Code Name	Products formerly Coffee Lake
Vertical Segment	Desktop
Processor Number	i7-8700K
Status	Launched
Launch Date	Q4'17
Lithography	14 nm
Instruction Set	64-bit

Performance

Number of Cores	6
Number of Threads	12
Processor Base Frequency	3.70 GHz
Max Turbo Frequency	4.70 GHz
Cache	12 MB Intel® Smart Cache
Bus Speed	8 GT/s
TDP	95 W

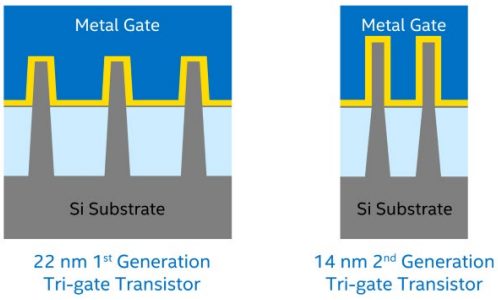
Memory Specifications

Max Memory Size	128 GB
Memory Types	DDR4-2666
Max of Memory Channels	2
Max Memory Bandwidth	41.6 GB/s
ECC Memory Supported	No

Expansion Options

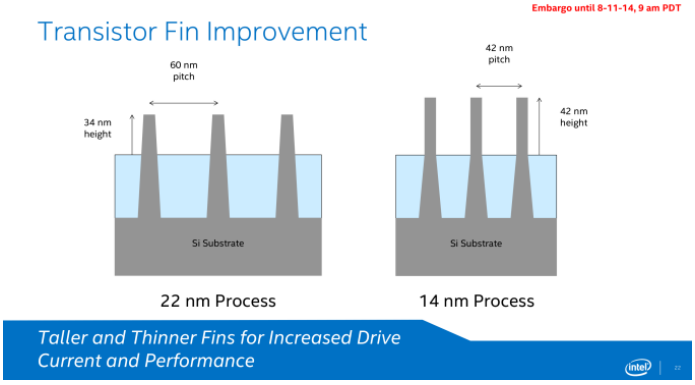
Scalability	1S Only
PCI Express Revision	3.0
PCI Express Configurations	Up to 1x16, 2x8, 1x8+2x4
Max of PCI Express Lanes	16

14 nm Chipset Architecture



	22 nm Node	14 nm Node	Scale
Transistor Fin Pitch	60	42	.70x
Transistor Gate Pitch	90	70	.78x
Interconnect Pitch	80 nm	52 nm	.65x

Intel® 14 nm technology provides good dimensional scaling from 22 nm.
The transistor fins are taller, thinner,
and more closely spaced for improved density and lower capacitance. Improved transistors req uire fewer fins, further improving density,
and the SRAM cell size is almost half the area of that in 22 nm.



Each one of these changes in turn improves the performance of the FinFETs in some wa y. The tighter density goes hand-in-
hand with 14nm's feature size reductions, while the taller, thinner fins allow for increased drive current and increased performance. Meanwhile by reducing the number of fins per transistor, Intel is able to improve on density once again while also reducing the transistor capacitance tha t results from those fins.

Pipeline in Intel core i7

CPU is 4-way superscalar, ~14 pipeline stages (P4 had 20!)

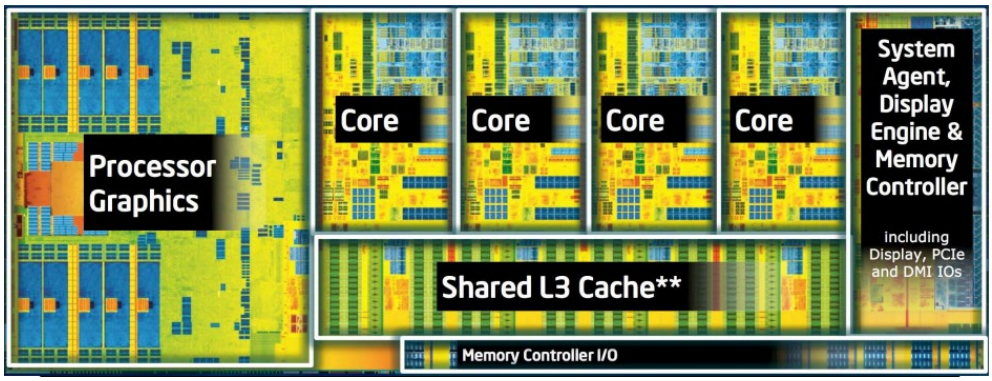
Superscalar picks from 96-Instruction window.

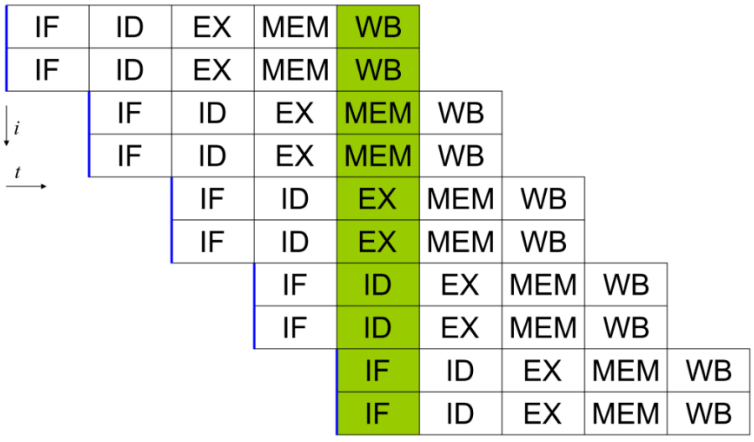
Register renaming to 168 registers.

Each chip has 4 cores

Symmetric Multithreading (2-way per core)

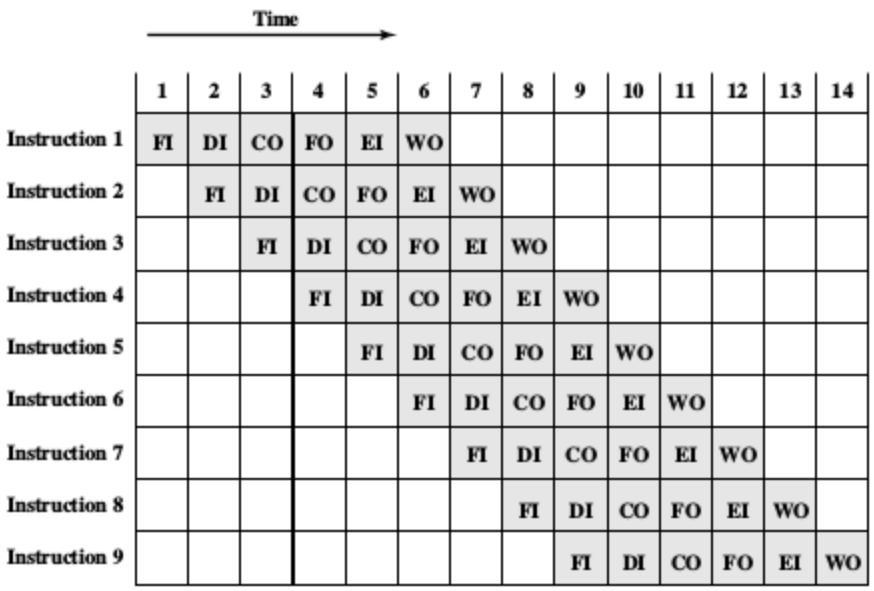
On-chip GPU





Parallelism Superscalar Pipeline in core i7

Simple superscalar pipeline. By fetching and dispatching two instructions at a time, a maximum of two instructions per cycle can be completed.



14 pipeline stages

1 cycle has 14 pipeline stages and 9 instruction can be 20-24 stages and instruction in one cycle(superscalar) 1 cycle do about Fetch instruction (FI) ,Decode instruction (DI), Calculate Operands (CO) , Fetch Operands (FO), Execute instruction(EI),, Write operand (WO) MEM = Memory access,

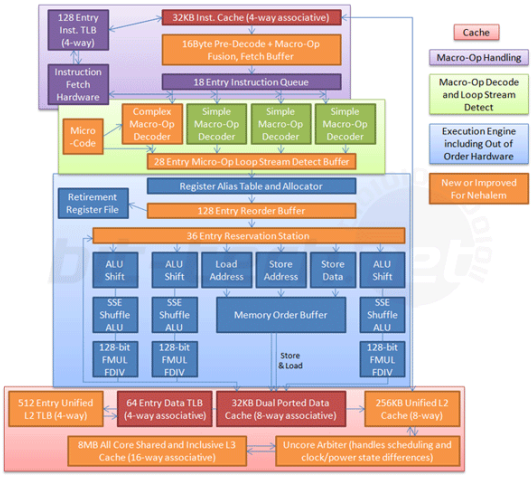
การทำงาน

First level cache and TLBs(translation lookaside buffer) remain the same at 32KB and 128 Entry (4-way associative) for data and instruction cache

Next Penryn and Conroe(processor name) pushed Macro-Op fusion as a performance feature - combining and condensing data before it even hits the Decoders

Next The Loop Stream Detection function in processor Pennyn doubled up as the Instruction and Queue hardware

Next Execution



สิ่งที่เปลี่ยนไป

Branch Prediction ดีขึ้น ->L2 Branch Predictor with better accuracy in large code sizes and an Advanced Renamed Return Stack Buffer (RSB) that removes branch mis predicts

Macro-Operation(in processor) it was limited to just 32-bit data making it - at best - fractionally useful in 64-bit systems that ran some 32-bit code. Nehalem updates the Macro-Ops fusion engine to finally handle and condense 64-bit Macro-Ops

LSD logic(in loop) so in effect respectively little extra logic was needed for this performance improvement. With Nehalem though, the LSD has been shifted down the pipeline a stage post Decode, requiring its own Entry Buffer hardware.

Execution engine improving performance Micro-Ops enter the Execution Engine they meet a new 28 Micro-Op buffer on Core 2 the engine can still only churn through four Micro-Ops a stage, so storing more from extra thread demands is necessary.

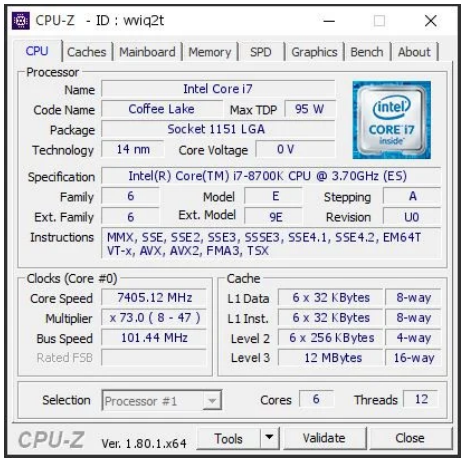
Buffer

-The Reorder Buffer has been made a third larger - up from 96 to 128 Entries , and the Reservation Station (which schedules operations to available Execution Units) has been given an extra four slots allowing 36 Entries also. This can execute up to six operations per clock cycle - three Memory Ops (a load, a store data and a store address) and three computational operations.

สรุป

Nehalem has been built to accommodate the pressures of a dual-thread environment through the pipeline.

Instruction Set



IA-64 uses new EPIC and Itanium processor

Performance core i7 8700k

IA-64 Architectural Features

Ex. Control & Data Speculation , Cache Control

64-bit Memory Access bit Memory Access

18 BILLION Giga Bytes accessible

Access granularity and alignment

Support for both Big and Little endian byte order

Memory hierarchy control

Register Stack

General Registers 0-31 are global to all procedures

Stacked registers begin at GR32 and are local to each procedure

Each procedure's register stack frame varies from 0 to 96 registers varies from 0 to 96 registers

Register Stack Engine (RSE)

A selection of the Core i7 integer instructions

1.Move

MOV DST,SRC	Move SRC to DST
PUSH SRC	Push SRC onto the stack
POP DST	Pop a word from the stack to DST
XCHG DS1,DS2	Exchange DS1 and DS2
LEA DST,SRC	Load effective addr of SRC into DST
CMOVcc DST,SRC	Conditional move

2.Arithmetic

ADD DST,SRC	Add SRC to DST
SUB DST,SRC	Subtract SRC from DST
MUL SRC	Multiply EAX by SRC (unsigned)
IMUL SRC	Multiply EAX by SRC (signed)
DIV SRC	Divide EDX:EAX by SRC (unsigned)
IDIV SRC	Divide EDX:EAX by SRC (signed)
ADC DST,SRC	Add SRC to DST, then add carry bit
SBB DST,SRC	Subtract SRC & carry from DST
INC DST	Add 1 to DST
DEC DST	Subtract 1 from DST
NEG DST	Negate DST (subtract it from 0)

3.Binary coded decimal

DAA	Decimal adjust
DAS	Decimal adjust for subtraction
AAA	ASCII adjust for addition
AAS	ASCII adjust for subtraction
AAM	ASCII adjust for multiplication
AAD	ASCII adjust for division

References

Ref: https://www.intel.com/content/dam/support/us/en/documents/processors/core/intel-core-i7-comparison-chart.pdf

Ref: https://www.cs.helsinki.fi/u/kerola/tikra/IA64-Architecture.pdf

Ref: http://www.it.uom.gr/teaching/tanenbaum/ed6/Chapter05-TheInstructionSetArchitectureLevel.pdf?fbclid=IwAR0syc5TXK5r5dROwq_o7AYWMmFFpSVlM7LAPjCcrtXsygglRAPQsjoMuYg

Ref: https://www.bit-tech.net/reviews/tech/cpus/intel-core-i7-nehalem-architecture-dive/5/

Ref: https://www.anandtech.com/show/8367/intels-14nm-technology-in-detail

Ref: https://class.ece.uw.edu/469/hauck/lectures/09_IntelHaswell.pdf

Ref: https://www.eit.lth.se/fileadmin/eit/courses/edt621/Rapporter/2016/Gregory_Austin_EDT621_Report.pdf