

SECTION G — PHYSICS

G06 COMPUTING; CALCULATING OR COUNTING

G06F ELECTRIC DIGITAL DATA PROCESSING (computer systems based on specific computational models G06N)

Note(s)

In this subclass, the following terms or expressions are used with the meaning indicated:

- "handling" includes processing or transporting of data;
- "data processing equipment" means an association of an electric digital data processor classifiable under group G06F 7/00, with one or more arrangements classifiable under groups G06F 1/00-G06F 5/00 and G06F 9/00-G06F 13/00.

Subclass index

DATA PROCESSING.....	7/00, 15/00-17/00
INPUT, OUTPUT; INTERCONNECTIONS BETWEEN FUNCTIONAL ELEMENTS.....	3/00, 13/00
ADDRESSING OR ALLOCATION.....	12/00
CONVERSION; SOFTWARE ENGINEERING; PROGRAMME CONTROL; ERROR DETECTION, MONITORING.....	5/00, 8/00, 9/00, 11/00
PATTERN RECOGNITION.....	18/00
DETAILS.....	1/00
SECURITY ARRANGEMENTS.....	21/00
COMPUTER-AIDED DESIGN [CAD].....	30/00
HANDLING NATURAL LANGUAGE DATA.....	40/00

1/00 Details not covered by groups G06F 3/00-G06F 13/00 and G06F 21/00 (architectures of general purpose stored program computers G06F 15/76) **[1, 2006.01]**

- 1/02 • Digital function generators **[1, 2006.01]**
- 1/025 • • for functions having two-valued amplitude, e.g. Walsh functions **[5, 2006.01]**
- 1/03 • • working, at least partly, by table look-up (G06F 1/025 takes precedence) **[5, 2006.01]**

Note(s) [5]

In order to be classified in this group, the table must contain function values of the desired or an intermediate function, not merely coefficients.

- 1/035 • • • Reduction of table size **[5, 2006.01]**
- 1/04 • Generating or distributing clock signals or signals derived directly therefrom **[1, 2006.01]**
- 1/06 • • Clock generators producing several clock signals **[5, 2006.01]**
- 1/08 • • Clock generators with changeable or programmable clock frequency **[5, 2006.01]**
- 1/10 • • Distribution of clock signals **[5, 2006.01]**
- 1/12 • • Synchronisation of different clock signals **[5, 2006.01]**
- 1/14 • • Time supervision arrangements, e.g. real time clock **[5, 2006.01]**
- 1/16 • Constructional details or arrangements **[5, 2006.01]**
- 1/18 • • Packaging or power distribution **[5, 2006.01]**
- 1/20 • • Cooling means **[5, 2006.01]**
- 1/22 • Means for limiting or controlling the pin/gate ratio **[5, 2006.01]**
- 1/24 • Resetting means **[5, 2006.01]**

- 1/26 • Power supply means, e.g. regulation thereof (for memories G11C) **[5, 2006.01]**
- 1/28 • • Supervision thereof, e.g. detecting power-supply failure by out of limits supervision **[5, 2006.01]**
- 1/30 • • Means for acting in the event of power-supply failure or interruption, e.g. power-supply fluctuations (for resetting only G06F 1/24) **[5, 2006.01]**
- 1/32 • • Means for saving power **[5, 2006.01, 2019.01]**
- 1/3203 • • • Power management, i.e. event-based initiation of a power-saving mode **[2019.01]**
- 1/3206 • • • • Monitoring of events, devices or parameters that trigger a change in power modality **[2019.01]**
- 1/3209 • • • • • Monitoring remote activity, e.g. over telephone lines or network connections **[2019.01]**
- 1/3212 • • • • • Monitoring battery levels, e.g. power saving mode being initiated when battery voltage goes below a certain level **[2019.01]**
- 1/3215 • • • • • Monitoring of peripheral devices **[2019.01]**
- 1/3218 • • • • • of display devices **[2019.01]**
- 1/3221 • • • • • of disk drive devices **[2019.01]**
- 1/3225 • • • • • of memory devices **[2019.01]**
- 1/3228 • • • • • Monitoring task completion, e.g. by use of idle timers, stop commands or wait commands **[2019.01]**
- 1/3231 • • • • • Monitoring the presence, absence or movement of users **[2019.01]**

- 1/3234 • • • • Power saving characterised by the action undertaken [2019.01]
- 1/3237 • • • • • by disabling clock generation or distribution [2019.01]
- 1/324 • • • • • by lowering clock frequency [2019.01]
- 1/3246 • • • • • by software initiated power-off [2019.01]
- 1/3287 • • • • • by switching off individual functional units in the computer system [2019.01]
- 1/329 • • • • • by task scheduling [2019.01]
- 1/3293 • • • • • by switching to a less power-consuming processor, e.g. sub-CPU [2019.01]
- 1/3296 • • • • • by lowering the supply or operating voltage [2019.01]

3/00 Input arrangements for transferring data to be processed into a form capable of being handled by the computer; Output arrangements for transferring data from processing unit to output unit, e.g. interface arrangements [1, 4, 2006.01]

- 3/01 • Input arrangements or combined input and output arrangements for interaction between user and computer (G06F 3/16 takes precedence) [2006.01]
- 3/02 • • Input arrangements using manually operated switches, e.g. using keyboards or dials [1, 3, 2006.01]
- 3/023 • • • Arrangements for converting discrete items of information into a coded form, e.g. arrangements for interpreting keyboard generated codes as alphanumeric codes, operand codes or instruction codes [3, 2006.01]
- 3/027 • • • • for insertion of the decimal point [3, 2006.01]
- 3/03 • • Arrangements for converting the position or the displacement of a member into a coded form [3, 2006.01]

Note(s) [2006.01]

In this group, the first place priority rule is applied, i.e. at each hierarchical level, in the absence of an indication to the contrary, classification is made in the first appropriate place.

- 3/033 • • • Pointing devices displaced or positioned by the user; Accessories therefor (digitisers characterised by the transducing means G06F 3/041) [3, 2006.01, 2013.01]
- 3/0338 • • • • with detection of limited linear or angular displacement of an operating part of the device from a neutral position, e.g. isotonic or isometric joysticks [2013.01]
- 3/0346 • • • • with detection of the device orientation or free movement in a 3D space, e.g. 3D mice, 6-DOF [six degrees of freedom] pointers using gyroscopes, accelerometers or tilt-sensors [2013.01]
- 3/0354 • • • • with detection of 2D relative movements between the device, or an operating part thereof, and a plane or surface, e.g. 2D mice, trackballs, pens or pucks [2013.01]
- 3/0362 • • • • with detection of 1D translations or rotations of an operating part of the device, e.g. scroll wheels, sliders, knobs, rollers or belts [2013.01]
- 3/037 • • • • using the raster scan of a cathode-ray tube [CRT] for detecting the position of the member, e.g. light pens cooperating with CRT monitors [3, 2006.01, 2013.01]

- 3/038 • • • • Control and interface arrangements therefor, e.g. drivers or device-embedded control circuitry [2006.01, 2013.01]
- 3/039 • • • • Accessories therefor, e.g. mouse pads [2006.01, 2013.01]
- 3/041 • • • Digitisers, e.g. for touch screens or touch pads, characterised by the transducing means [2006.01]
- 3/042 • • • • by opto-electronic means [2006.01]
- 3/043 • • • • using propagating acoustic waves [2006.01]
- 3/044 • • • • by capacitive means [2006.01]
- 3/045 • • • • using resistive elements, e.g. a single continuous surface or two parallel surfaces put in contact [2006.01]
- 3/046 • • • • by electromagnetic means [2006.01]
- 3/047 • • • • using sets of wires, e.g. crossed wires [2006.01]
- 3/048 • • Interaction techniques based on graphical user interfaces [GUI] [2006.01, 2013.01]

Note(s) [2013.01]

This group covers subject matter where the focus is placed on the way the user can interact with the displayed data. The mere presence of a standard GUI in the context of the disclosure of a specific software application or a specific device capable of processing data related to its specific function, should in general be classified in the appropriate subclasses related to those software applications or specific devices.

- 3/0481 • • • based on specific properties of the displayed interaction object or a metaphor-based environment, e.g. interaction with desktop elements like windows or icons, or assisted by a cursor's changing behaviour or appearance [2013.01, 2022.01]
- 3/04812• • • • Interaction techniques based on cursor appearance or behaviour, e.g. being affected by the presence of displayed objects [2022.01]
- 3/04815• • • • Interaction with a metaphor-based environment or interaction object displayed as three-dimensional, e.g. changing the user viewpoint with respect to the environment or object [2022.01]
- 3/04817• • • • using icons (graphical or visual programming using iconic symbols G06F 8/34) [2022.01]
- 3/0482 • • • • Interaction with lists of selectable items, e.g. menus [2013.01]
- 3/0483 • • • • Interaction with page-structured environments, e.g. book metaphor [2013.01]
- 3/0484 • • • • for the control of specific functions or operations, e.g. selecting or manipulating an object, an image or a displayed text element, setting a parameter value or selecting a range [2013.01, 2022.01]
- 3/04842• • • • Selection of displayed objects or displayed text elements (G06F 3/0482 takes precedence) [2022.01]
- 3/04845• • • • for image manipulation, e.g. dragging, rotation, expansion or change of colour [2022.01]
- 3/04847• • • • Interaction techniques to control parameter settings, e.g. interaction with sliders or dials [2022.01]
- 3/0485 • • • • Scrolling or panning [2013.01, 2022.01]
- 3/04855• • • • • Interaction with scrollbars [2022.01]
- 3/0486 • • • • Drag-and-drop [2013.01]

- 3/0487 • • • using specific features provided by the input device, e.g. functions controlled by the rotation of a mouse with dual sensing arrangements, or of the nature of the input device, e.g. tap gestures based on pressure sensed by a digitiser [2013.01]
- 3/0488 • • • using a touch-screen or digitiser, e.g. input of commands through traced gestures [2013.01, 2022.01]
- 3/04883 • • • for inputting data by handwriting, e.g. gesture or text [2022.01]
- 3/04886 • • • by partitioning the display area of the touch-screen or the surface of the digitising tablet into independently controllable areas, e.g. virtual keyboards or menus [2022.01]
- 3/0489 • • • using dedicated keyboard keys or combinations thereof [2013.01, 2022.01]
- 3/04892 • • • Arrangements for controlling cursor position based on codes indicative of cursor displacements from one discrete location to another, e.g. using cursor control keys associated to different directions or using the tab key (arrangements for controlling cursor position based on coordinate signals G06F 3/038) [2022.01]
- 3/04895 • • • Guidance during keyboard input operation, e.g. prompting [2022.01]
- 3/05 • Digital input using the sampling of an analogue quantity at regular intervals of time [1, 2006.01]
- 3/06 • Digital input from, or digital output to, record carriers [1, 2006.01]
- 3/08 • • from or to individual record carriers, e.g. punched card [1, 2006.01]
- 3/09 • Digital output to typewriters [3, 2006.01]
- 3/12 • Digital output to print unit [1, 2006.01]
- 3/13 • Digital output to plotter [3, 2006.01]
- 3/14 • Digital output to display device [1, 2006.01]
- 3/147 • • using display panels [3, 2006.01]
- 3/153 • • using cathode-ray tubes [3, 2006.01]
- 3/16 • Sound input; Sound output (speech processing G10L) [1, 2006.01]
- 3/18 • Digital input from automatic curve follower [3, 2006.01]
- 5/00 Methods or arrangements for data conversion without changing the order or content of the data handled [1, 4, 2006.01]**
- 5/01 • for shifting, e.g. justifying, scaling, normalising [5, 2006.01]
- 5/06 • for changing the speed of data flow, i.e. speed regularising [1, 2006.01]
- 5/08 • • having a sequence of storage locations, the intermediate ones not being accessible for either enqueue or dequeue operations, e.g. using a shift register [2006.01]
- 5/10 • • having a sequence of storage locations each being individually accessible for both enqueue and dequeue operations, e.g. using random access memory [2006.01]
- 5/12 • • • Means for monitoring the fill level; Means for resolving contention, i.e. conflicts between simultaneous enqueue and dequeue operations [2006.01]
- 5/14 • • • for overflow or underflow handling, e.g. full or empty flags [2006.01]
- 5/16 • • Multiplexed systems, i.e. using two or more similar devices which are alternately accessed for enqueue and dequeue operations, e.g. ping-pong buffers [2006.01]
- 7/00 Methods or arrangements for processing data by operating upon the order or content of the data handled (logic circuits H03K 19/00) [1, 2006.01]**
- 7/02 • Comparing digital values (G06F 7/06, G06F 7/38 take precedence) [1, 2006.01]
- 7/04 • • Identity comparison, i.e. for like or unlike values [1, 2006.01]
- 7/06 • Arrangements for sorting, selecting, merging, or comparing data on individual record carriers [1, 2006.01]
- 7/08 • • Sorting, i.e. grouping record carriers in numerical or other ordered sequence according to the classification of at least some of the information they carry (by merging two or more sets of carriers in ordered sequence G06F 7/16) [1, 2006.01]
- 7/10 • • Selecting, i.e. obtaining data of one kind from those record carriers which are identifiable by data of a second kind from a mass of ordered or randomly-distributed record carriers [1, 2006.01]
- 7/12 • • • with provision for printing-out a list of selected items [1, 2006.01]
- 7/14 • • Merging, i.e. combining at least two sets of record carriers each arranged in the same ordered sequence to produce a single set having the same ordered sequence [1, 2006.01]
- 7/16 • • • Combined merging and sorting [1, 2006.01]
- 7/20 • • Comparing separate sets of record carriers arranged in the same sequence to determine whether at least some of the data in one set is identical with that in the other set or sets [1, 2006.01]
- 7/22 • Arrangements for sorting or merging computer data on continuous record carriers, e.g. tape, drum, disc [1, 2006.01]
- 7/24 • • Sorting, i.e. extracting data from one or more carriers, re-arranging the data in numerical or other ordered sequence, and re-recording the sorted data on the original carrier or on a different carrier or set of carriers (G06F 7/36 takes precedence) [1, 2006.01]
- 7/26 • • • the sorted data being recorded on the original record carrier within the same space in which the data had been recorded prior to their sorting, without using intermediate storage [1, 2006.01]
- 7/32 • • Merging, i.e. combining data contained in ordered sequence on at least two record carriers to produce a single carrier or set of carriers having all the original data in the ordered sequence (G06F 7/36 takes precedence) [1, 2006.01]
- 7/36 • • Combined merging and sorting [1, 2006.01]
- 7/38 • Methods or arrangements for performing computations using exclusively denominational number representation, e.g. using binary, ternary, decimal representation [1, 3, 2006.01]
- 7/40 • • using contact-making devices, e.g. electromagnetic relay (G06F 7/46 takes precedence) [1, 2006.01]
- 7/42 • • • Adding; Subtracting [1, 2006.01]
- 7/44 • • • Multiplying; Dividing [1, 2006.01]
- 7/46 • • using electromechanical counter-type accumulators [1, 2006.01]

- 7/48 • • using non-contact-making devices, e.g. tube, solid state device; using unspecified devices [1, 3, 2006.01]
- 7/483 • • • Computations with numbers represented by a non-linear combination of denominational numbers, e.g. rational numbers, logarithmic number system or floating-point numbers [2006.01]
- 7/485 • • • Adding; Subtracting [2006.01]
- 7/487 • • • Multiplying; Dividing [2006.01]
- 7/49 • • • Computations with a radix, other than binary, 8, 16 or decimal, e.g. ternary, negative or imaginary radices, mixed radix [3, 2006.01]
- 7/491 • • • Computations with decimal numbers [2006.01]
- 7/492 • • • • using a binary weighted representation within each denomination [2006.01]
- 7/493 • • • • • the representation being the natural binary coded representation, i.e. 8421-code [2006.01]
- 7/494 • • • • • Adding; Subtracting [2006.01]
- 7/495 • • • • • • in digit-serial fashion, i.e. having a single digit-handling circuit treating all denominations after each other [2006.01]
- 7/496 • • • • • Multiplying; Dividing [2006.01]
- 7/498 • • • • using counter-type accumulators [2006.01]
- 7/499 • • • Denomination or exception handling, e.g. rounding or overflow [2006.01]
- 7/50 • • • Adding; Subtracting (G06F 7/483-G06F 7/491, G06F 7/544-G06F 7/556 take precedence) [1, 3, 2006.01]
- 7/501 • • • • Half or full adders, i.e. basic adder cells for one denomination [2006.01]
- 7/502 • • • • • Half adders; Full adders consisting of two cascaded half adders [2006.01]
- 7/503 • • • • • using carry switching, i.e. the incoming carry being connected directly, or only via an inverter, to the carry output under control of a carry propagate signal [2006.01]
- 7/504 • • • • • in bit-serial fashion, i.e. having a single digit-handling circuit treating all denominations after each other [2006.01]
- 7/505 • • • • • in bit-parallel fashion, i.e. having a different digit-handling circuit for each denomination [2006.01]
- 7/506 • • • • • • with simultaneous carry generation for, or propagation over, two or more stages [2006.01]
- 7/507 • • • • • • using selection between two conditionally calculated carry or sum values [2006.01]
- 7/508 • • • • • • using carry look-ahead circuits [2006.01]
- 7/509 • • • • • • for multiple operands, e.g. digital integrators [2006.01]
- 7/52 • • • Multiplying; Dividing (G06F 7/483-G06F 7/491, G06F 7/544-G06F 7/556 take precedence) [1, 3, 2006.01]
- 7/523 • • • • Multiplying only [2006.01]
- 7/525 • • • • • in serial-serial fashion, i.e. both operands being entered serially (G06F 7/533 takes precedence) [2006.01]
- 7/527 • • • • • in serial-parallel fashion, i.e. one operand being entered serially and the other in parallel (G06F 7/533 takes precedence) [2006.01]
- 7/53 • • • • • in parallel-parallel fashion, i.e. both operands being entered in parallel (G06F 7/533 takes precedence) [2006.01]
- 7/533 • • • • • Reduction of the number of iteration steps or stages, e.g. using the Booth algorithm, log-sum, odd-even [2006.01]
- 7/535 • • • • Dividing only [2006.01]
- 7/537 • • • • • Reduction of the number of iteration steps or stages, e.g. using the Sweeney-Robertson-Tocher [SRT] algorithm [2006.01]
- 7/544 • • • • for evaluating functions by calculation [3, 2006.01]
- 7/548 • • • • Trigonometric functions; Co-ordinate transformations [3, 2006.01]
- 7/552 • • • • Powers or roots [3, 2006.01]
- 7/556 • • • • Logarithmic or exponential functions [3, 2006.01]
- 7/57 • • • Arithmetic logic units [ALU], i.e. arrangements or devices for performing two or more of the operations covered by groups G06F 7/483-G06F 7/556 or for performing logical operations [2006.01]
- 7/575 • • • • Basic arithmetic logic units, i.e. devices selectable to perform either addition, subtraction or one of several logical operations, using, at least partially, the same circuitry [2006.01]
- 7/58 • Random or pseudo-random number generators [3, 2006.01]
- 7/60 • Methods or arrangements for performing computations using a digital non-denominational number representation, i.e. number representation without radix; Computing devices using combinations of denominational and non-denominational quantity representations [3, 2006.01]
- 7/62 • • Performing operations exclusively by counting total number of pulses [3, 2006.01]
- 7/64 • • Digital differential analysers, i.e. computing devices for differentiation, integration or solving differential or integral equations, using pulses representing increments; Other incremental computing devices for solving difference equations (G06F 7/70 takes precedence; differential analysers using hybrid computing techniques G06J 1/02) [3, 2006.01]
- 7/66 • • • wherein pulses represent unitary increments only [3, 2006.01]
- 7/68 • • using pulse rate multipliers or dividers (G06F 7/70 takes precedence) [3, 2006.01]
- 7/70 • • using stochastic pulse trains, i.e. randomly occurring pulses the average pulse rates of which represent numbers [3, 2006.01]
- 7/72 • • using residue arithmetic [3, 2006.01]
- 7/74 • Selecting or encoding within a word the position of one or more bits having a specified value, e.g. most or least significant one or zero detection, priority encoders [2006.01]
- 7/76 • Arrangements for rearranging, permuting or selecting data according to predetermined rules, independently of the content of the data [2006.01]
- 7/78 • • for changing the order of data flow, e.g. matrix transposition or LIFO buffers; Overflow or underflow handling therefor [2006.01]

- 8/00 Arrangements for software engineering** (testing or debugging G06F 11/36; administrative, planning or organisation aspects of software project management G06Q 10/06) **[2018.01]**
- 8/10 • Requirements analysis; Specification techniques **[2018.01]**
- 8/20 • Software design **[2018.01]**
- 8/30 • Creation or generation of source code **[2018.01]**
- 8/33 • • Intelligent editors **[2018.01]**
- 8/34 • • Graphical or visual programming **[2018.01]**
- 8/35 • • model driven **[2018.01]**
- 8/36 • • Software reuse **[2018.01]**
- 8/38 • • for implementing user interfaces **[2018.01]**
- 8/40 • Transformation of program code **[2018.01]**
- 8/41 • • Compilation **[2018.01]**
- 8/51 • • Source to source **[2018.01]**
- 8/52 • • Binary to binary **[2018.01]**
- 8/53 • • Decompilation; Disassembly **[2018.01]**
- 8/54 • • Link editing before load time **[2018.01]**
- 8/60 • Software deployment **[2018.01]**
- 8/61 • • Installation **[2018.01]**
- 8/65 • • Updates (security arrangements therefor G06F 21/57) **[2018.01]**
- 8/654 • • • using techniques specially adapted for alterable solid state memories, e.g. for EEPROM or flash memories **[2018.01]**
- 8/656 • • • while running **[2018.01]**
- 8/658 • • • Incremental updates; Differential updates **[2018.01]**
- 8/70 • Software maintenance or management **[2018.01]**
- 8/71 • • Version control (security arrangements therefor G06F 21/57); Configuration management **[2018.01]**
- 8/72 • • Code refactoring **[2018.01]**
- 8/73 • • Program documentation **[2018.01]**
- 8/74 • • Reverse engineering; Extracting design information from source code **[2018.01]**
- 8/75 • • Structural analysis for program understanding **[2018.01]**
- 8/76 • • Adapting program code to run in a different environment; Porting **[2018.01]**
- 8/77 • • Software metrics **[2018.01]**
- 9/00 Arrangements for program control, e.g. control units** (program control for peripheral devices G06F 13/10) **[1, 4, 2006.01, 2018.01]**
- 9/02 • using wired connections, e.g. plugboards **[1, 2006.01]**
- 9/04 • using record carriers containing only program instructions (G06F 9/06 takes precedence) **[1, 2006.01]**
- 9/06 • using stored programs, i.e. using an internal store of processing equipment to receive or retain programs **[1, 2006.01]**
- 9/22 • • Microcontrol or microprogram arrangements **[3, 2006.01]**
- 9/24 • • • Loading of the microprogram **[3, 2006.01]**
- 9/26 • • • Address formation of the next microinstruction (G06F 9/28 takes precedence) **[3, 2006.01]**
- 9/28 • • • Enhancement of operational speed, e.g. by using several microcontrol devices operating in parallel **[3, 2006.01]**
- 9/30 • • Arrangements for executing machine instructions, e.g. instruction decode (for executing microinstructions G06F 9/22) **[3, 2006.01, 2018.01]**
- 9/302 • • • Controlling the executing of arithmetic operations **[5, 2006.01, 2018.01]**
- 9/305 • • • Controlling the executing of logical operations **[5, 2006.01, 2018.01]**
- 9/308 • • • Controlling single bit operations (G06F 9/305 takes precedence) **[5, 2006.01, 2018.01]**
- 9/312 • • • Controlling loading, storing or clearing operations **[5, 2006.01, 2018.01]**
- 9/315 • • • Controlling moving, shifting or rotation operations **[5, 2006.01, 2018.01]**
- 9/318 • • • with operation extension or modification **[5, 2006.01, 2018.01]**
- 9/32 • • • Address formation of the next instruction, e.g. by incrementing the instruction counter (G06F 9/38 takes precedence) **[3, 2006.01, 2018.01]**
- 9/34 • • • Addressing or accessing the instruction operand or the result **[3, 5, 2006.01, 2018.01]**
- 9/345 • • • • of multiple operands or results **[5, 2006.01, 2018.01]**
- 9/35 • • • • Indirect addressing **[5, 2006.01, 2018.01]**
- 9/355 • • • • Indexed addressing **[5, 2006.01, 2018.01]**
- 9/38 • • • Concurrent instruction execution, e.g. pipeline or look ahead **[3, 2006.01, 2018.01]**
- 9/44 • • Arrangements for executing specific programs **[3, 2006.01, 2018.01]**
- 9/4401 • • • Bootstrapping (security arrangements therefor G06F 21/57) **[2018.01]**
- 9/445 • • • Program loading or initiating (bootstrapping G06F 9/4401; security arrangements for program loading or initiating G06F 21/57) **[5, 2006.01, 2018.01]**
- 9/448 • • • Execution paradigms, e.g. implementations of programming paradigms **[2018.01]**
- 9/451 • • • Execution arrangements for user interfaces **[2018.01]**
- 9/455 • • • Emulation; Interpretation; Software simulation, e.g. virtualisation or emulation of application or operating system execution engines **[5, 2006.01, 2018.01]**
- 9/46 • • Multiprogramming arrangements **[3, 2006.01]**
- 9/48 • • • Program initiating; Program switching, e.g. by interrupt **[7, 2006.01]**
- 9/50 • • • Allocation of resources, e.g. of the central processing unit [CPU] **[7, 2006.01]**
- 9/52 • • • Program synchronisation; Mutual exclusion, e.g. by means of semaphores **[7, 2006.01]**
- 9/54 • • • Interprogram communication **[7, 2006.01]**
- 11/00 Error detection; Error correction; Monitoring** (error detection, correction or monitoring in information storage based on relative movement between record carrier and transducer G11B 20/18; monitoring, i.e. supervising the progress of recording or reproducing G11B 27/36; in static stores G11C 29/00) **[1, 4, 2006.01]**
- 11/07 • Responding to the occurrence of a fault, e.g. fault tolerance **[7, 2006.01]**
- 11/08 • • Error detection or correction by redundancy in data representation, e.g. by using checking codes **[1, 2006.01]**
- 11/10 • • • Adding special bits or symbols to the coded information, e.g. parity check, casting out nines or elevens **[1, 2006.01]**
- 11/14 • • Error detection or correction of the data by redundancy in operation, e.g. by using different operation sequences leading to the same result (G06F 11/16 takes precedence) **[3, 2006.01]**

- 11/16 • • Error detection or correction of the data by redundancy in hardware [3, 2006.01]
- 11/18 • • • using passive fault-masking of the redundant circuits, e.g. by quadding or by majority decision circuits [3, 2006.01]
- 11/20 • • • using active fault-masking, e.g. by switching out faulty elements or by switching in spare elements [3, 2006.01]
- 11/22 • Detection or location of defective computer hardware by testing during standby operation or during idle time, e.g. start-up testing [3, 2006.01]
- 11/24 • • Marginal testing [3, 2006.01]
- 11/25 • • Testing of logic operation, e.g. by logic analysers [6, 2006.01]
- 11/26 • • Functional testing [3, 2006.01]
- 11/263 • • • Generation of test inputs, e.g. test vectors, patterns or sequences [6, 2006.01]
- 11/267 • • • Reconfiguring circuits for testing, e.g. LSSD, partitioning [6, 2006.01]
- 11/27 • • • Built-in tests [6, 2006.01]
- 11/273 • • • Tester hardware, i.e. output processing circuits [6, 2006.01]
- 11/277 • • • with comparison between actual response and known fault-free response [6, 2006.01]
- 11/28 • by checking the correct order of processing (G06F 11/07, G06F 11/22 take precedence) [3, 2006.01]
- 11/30 • Monitoring [3, 2006.01]
- 11/32 • • with visual indication of the functioning of the machine [3, 2006.01]
- 11/34 • • Recording or statistical evaluation of computer activity, e.g. of down time, of input/output operation [3, 2006.01]
- 11/36 • *Prevention of errors by analysis, debugging or testing of software [7, 2006.01, 2025.01]*
- 11/3604 • • *Analysis of software for verifying properties of programs (testing of software G06F 11/3668) [2025.01]*
- 11/362 • • *Debugging of software [2025.01]*
- 11/3668 • • *Testing of software [2025.01]*
- 11/3698 • • *Environments for analysis, debugging or testing of software [2025.01]*
- 12/00 Accessing, addressing or allocating within memory systems or architectures** (digital input from, or digital output to record carriers, e.g. to disk storage units, G06F 3/06) [4, 5, 2006.01]
- 12/02 • Addressing or allocation; Relocation (program address sequencing G06F 9/00; arrangements for selecting an address in a digital store G11C 8/00) [4, 2006.01]
- 12/04 • • Addressing variable-length words or parts of words [4, 2006.01]
- 12/06 • • Addressing a physical block of locations, e.g. base addressing, module addressing, address space extension, memory dedication (G06F 12/08 takes precedence) [4, 2006.01]
- 12/08 • • in hierarchically structured memory systems, e.g. virtual memory systems [4, 2006.01, 2016.01]
- 12/0802 • • • Addressing of a memory level in which the access to the desired data or data block requires associative addressing means, e.g. caches [2016.01]
- 12/0804 • • • with main memory updating (G06F 12/0806 takes precedence) [2016.01]
- 12/0806 • • • Multiuser, multiprocessor or multiprocessing cache systems [2016.01]
- 12/0808 • • • • with cache invalidating means (G06F 12/0815 takes precedence) [2016.01]
- 12/0811 • • • • with multilevel cache hierarchies [2016.01]
- 12/0813 • • • • with a network or matrix configuration [2016.01]
- 12/0815 • • • • Cache consistency protocols [2016.01]
- 12/0817 • • • • • using directory methods [2016.01]
- 12/0831 • • • • • using a bus scheme, e.g. with bus monitoring or watching means [2016.01]
- 12/0837 • • • • • with software control, e.g. non-cacheable data [2016.01]
- 12/084 • • • • • with a shared cache [2016.01]
- 12/0842 • • • • • for multiprocessing or multitasking [2016.01]
- 12/0844 • • • • Multiple simultaneous or quasi-simultaneous cache accessing [2016.01]
- 12/0846 • • • • • Cache with multiple tag or data arrays being simultaneously accessible [2016.01]
- 12/0853 • • • • • Cache with multiport tag or data arrays [2016.01]
- 12/0855 • • • • • Overlapped cache accessing, e.g. pipeline (G06F 12/0846 takes precedence) [2016.01]
- 12/0862 • • • • • with prefetch [2016.01]
- 12/0864 • • • • • using pseudo-associative means, e.g. set-associative or hashing [2016.01]
- 12/0866 • • • • • for peripheral storage systems, e.g. disk cache [2016.01]
- 12/0868 • • • • • Data transfer between cache memory and other subsystems, e.g. storage devices or host systems [2016.01]
- 12/0871 • • • • • Allocation or management of cache space [2016.01]
- 12/0873 • • • • • Mapping of cache memory to specific storage devices or parts thereof [2016.01]
- 12/0875 • • • • • with dedicated cache, e.g. instruction or stack [2016.01]
- 12/0877 • • • • • Cache access modes [2016.01]
- 12/0879 • • • • • Burst mode [2016.01]
- 12/0882 • • • • • Page mode [2016.01]
- 12/0884 • • • • • Parallel mode, e.g. in parallel with main memory or CPU [2016.01]
- 12/0886 • • • • • Variable-length word access [2016.01]
- 12/0888 • • • • • using selective caching, e.g. bypass [2016.01]
- 12/0891 • • • • • using clearing, invalidating or resetting means [2016.01]
- 12/0893 • • • • • Caches characterised by their organisation or structure [2016.01]
- 12/0895 • • • • • of parts of caches, e.g. directory or tag array [2016.01]
- 12/0897 • • • • • with two or more cache hierarchy levels (with multilevel cache hierarchies G06F 12/0811) [2016.01]
- 12/10 • • • Address translation [4, 2006.01, 2016.01]
- 12/1009 • • • • using page tables, e.g. page table structures [2016.01]
- 12/1018 • • • • • involving hashing techniques, e.g. inverted page tables [2016.01]
- 12/1027 • • • • • using associative or pseudo-associative address translation means, e.g. translation look-aside buffer [TLB] [2016.01]

- 12/1036 • • • • • for multiple virtual address spaces, e.g. segmentation (G06F 12/1045 takes precedence) [2016.01]
- 12/1045 • • • • • associated with a data cache [2016.01]
- 12/1072 • • • • • Decentralised address translation, e.g. in distributed shared memory systems [2016.01]
- 12/1081 • • • • • for peripheral access to main memory, e.g. direct memory access [DMA] [2016.01]
- 12/109 • • • • • for multiple virtual address spaces, e.g. segmentation (G06F 12/1036 takes precedence) [2016.01]
- 12/12 • • • • Replacement control [4, 2006.01, 2016.01]
- 12/121 • • • • • using replacement algorithms [2016.01]
- 12/122 • • • • • of the least frequently used [LFU] type, e.g. with individual count value [2016.01]
- 12/123 • • • • • with age lists, e.g. queue, most recently used [MRU] list or least recently used [LRU] list [2016.01]
- 12/126 • • • • • with special data handling, e.g. priority of data or instructions, handling errors or pinning [2016.01]
- 12/127 • • • • • using additional replacement algorithms [2016.01]
- 12/128 • • • • • adapted to multidimensional cache systems, e.g. set-associative, multicache, multiset or multilevel [2016.01]
- 12/14 • Protection against unauthorised use of memory [4, 2006.01]
- 12/16 • Protection against loss of memory contents [4, 2006.01]
- 13/00 Interconnection of, or transfer of information or other signals between, memories, input/output devices or central processing units** (interface circuits for specific input/output devices G06F 3/00; multi-processor systems G06F 15/16) [1, 4, 2006.01]
- 13/10 • Program control for peripheral devices (G06F 13/14-G06F 13/42 take precedence) [4, 2006.01]
- 13/12 • • using hardware independent of the central processor, e.g. channel or peripheral processor [4, 2006.01]
- 13/14 • Handling requests for interconnection or transfer [4, 2006.01]
- 13/16 • • for access to memory bus (G06F 13/28 takes precedence) [4, 2006.01]
- 13/18 • • • with priority control [4, 2006.01]
- 13/20 • • for access to input/output bus [4, 2006.01]
- 13/22 • • • using successive scanning, e.g. polling (G06F 13/24 takes precedence) [4, 2006.01]
- 13/24 • • • using interrupt (G06F 13/32 takes precedence) [4, 2006.01]
- 13/26 • • • • with priority control [4, 2006.01]
- 13/28 • • • using burst mode transfer, e.g. direct memory access, cycle steal (G06F 13/32 takes precedence) [4, 2006.01]
- 13/30 • • • • with priority control [4, 2006.01]
- 13/32 • • • using combination of interrupt and burst mode transfer [4, 2006.01]
- 13/34 • • • • with priority control [4, 2006.01]
- 13/36 • • for access to common bus or bus system [4, 2006.01]
- 13/362 • • • with centralised access control [5, 2006.01]
- 13/364 • • • • using independent requests or grants, e.g. using separated request and grant lines [5, 2006.01]
- 13/366 • • • • using a centralised polling arbiter [5, 2006.01]
- 13/368 • • • • with decentralised access control [5, 2006.01]
- 13/37 • • • • using a physical-position-dependent priority, e.g. daisy chain, round robin or token passing [5, 2006.01]
- 13/372 • • • • using a time-dependent priority, e.g. individually loaded time counters or time slot [5, 2006.01]
- 13/374 • • • • using a self-select method with individual priority code comparator [5, 2006.01]
- 13/376 • • • • using a contention resolving method, e.g. collision detection, collision avoidance [5, 2006.01]
- 13/378 • • • • using a parallel poll method [5, 2006.01]
- 13/38 • Information transfer, e.g. on bus (G06F 13/14 takes precedence) [4, 2006.01]
- 13/40 • • Bus structure [4, 2006.01]
- 13/42 • • Bus transfer protocol, e.g. handshake; Synchronisation [4, 2006.01]
- 15/00 Digital computers in general** (details G06F 1/00-G06F 13/00); **Data processing equipment in general** [1, 2006.01]
- 15/02 • manually operated with input through keyboard and computation using a built-in program, e.g. pocket calculators [1, 2006.01]
- 15/04 • programmed simultaneously with the introduction of data to be processed, e.g. on the same record carrier [1, 2006.01]
- 15/08 • using a plugboard for programming [1, 5, 2006.01]
- 15/10 • • Tabulators [1, 5, 2006.01]
- 15/12 • • • having provision for both printed and punched output [1, 5, 2006.01]
- 15/14 • • Calculating-punches [1, 5, 2006.01]
- 15/16 • Combinations of two or more digital computers each having at least an arithmetic unit, a program unit and a register, e.g. for a simultaneous processing of several programs [1, 2006.01]
- 15/163 • • Interprocessor communication [6, 2006.01]
- 15/167 • • • using a common memory, e.g. mailbox [6, 2006.01]
- 15/17 • • • using an input/output type connection, e.g. channel, I/O port [6, 2006.01]
- 15/173 • • • using an interconnection network, e.g. matrix, shuffle, pyramid, star or snowflake [6, 2006.01]
- 15/177 • • Initialisation or configuration control (configuration control for monitoring, testing or in case of failure G06F 11/00) [6, 2006.01]
- 15/76 • Architectures of general purpose stored program computers (with program plugboard G06F 15/08; multicomputers G06F 15/16) [5, 6, 2006.01]
- 15/78 • • comprising a single central processing unit [5, 2006.01]
- 15/80 • • comprising an array of processing units with common control, e.g. single instruction multiple data processors (G06F 15/82 takes precedence) [5, 2006.01]
- 15/82 • • data or demand driven [5, 2006.01]
- 16/00 Information retrieval; Database structures therefor; File system structures therefor** [2019.01]
- 16/10 • File systems; File servers [2019.01]
- 16/11 • • File system administration, e.g. details of archiving or snapshots (file system backup G06F 11/14) [2019.01]

G06F

- 16/13 • • File access structures, e.g. distributed indices (arrangements of input from, or output to, record carriers G06F 3/06) [2019.01]
- 16/14 • • Details of searching files based on file metadata [2019.01]
- 16/16 • • File or folder operations, e.g. details of user interfaces specifically adapted to file systems [2019.01]
- 16/17 • • Details of further file system functions [2019.01]
- 16/172 • • • Caching, prefetching or hoarding of files [2019.01]
- 16/174 • • • Redundancy elimination performed by the file system (management of the data involved in backup or backup restore using de-duplication of the data G06F 11/14) [2019.01]
- 16/176 • • • Support for shared access to files; File sharing support [2019.01]
- 16/178 • • • Techniques for file synchronisation in file systems [2019.01]
- 16/18 • • File system types [2019.01]
- 16/182 • • • Distributed file systems [2019.01]
- 16/185 • • • Hierarchical storage management [HSM] systems, e.g. file migration or policies thereof (details of archiving G06F 16/11) [2019.01]
- 16/188 • • • Virtual file systems [2019.01]
- 16/20 • of structured data, e.g. relational data [2019.01]
- 16/21 • • Design, administration or maintenance of databases [2019.01]
- 16/215 • • • Improving data quality; Data cleansing, e.g. de-duplication, removing invalid entries or correcting typographical errors [2019.01]
- 16/22 • • Indexing; Data structures therefor; Storage structures [2019.01]
- 16/23 • • Updating [2019.01]
- 16/24 • • Querying [2019.01]
- 16/242 • • • Query formulation [2019.01]
- 16/245 • • • Query processing [2019.01]
- 16/2452 • • • • Query translation [2019.01]
- 16/2453 • • • • Query optimisation [2019.01]
- 16/2455 • • • • Query execution [2019.01]
- 16/2457 • • • • with adaptation to user needs [2019.01]
- 16/2458 • • • • Special types of queries, e.g. statistical queries, fuzzy queries or distributed queries [2019.01]
- 16/248 • • • Presentation of query results [2019.01]
- 16/25 • • Integrating or interfacing systems involving database management systems [2019.01]
- 16/26 • • Visual data mining; Browsing structured data [2019.01]
- 16/27 • • Replication, distribution or synchronisation of data between databases or within a distributed database system; Distributed database system architectures therefor [2019.01]
- 16/28 • • Databases characterised by their database models, e.g. relational or object models [2019.01]
- 16/29 • • Geographical information databases [2019.01]
- 16/30 • of unstructured textual data (document management systems G06F 16/93) [2019.01]

Note(s) [2019.01]

In groups G06F 16/30-G06F 16/36, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/38-G06F 16/387.

- 16/31 • • Indexing; Data structures therefor; Storage structures [2019.01]

- 16/33 • • Querying [2019.01, 2025.01]
- 16/332 • • • Query formulation [2019.01, 2025.01]
- 16/3329 • • • • Natural language query formulation [2025.01]
- 16/3331 • • • Query processing [2025.01]
- 16/3332 • • • • Query translation [2025.01]
- 16/334 • • • • Query execution (filtering based on additional data G06F 16/335) [2025.01]
- 16/3349 • • • • Reuse of stored results of previous queries [2025.01]
- 16/335 • • • Filtering based on additional data, e.g. user or group profiles (filtering in web context G06F 16/9535, G06F 16/9536) [2019.01]
- 16/338 • • • Presentation of query results [2019.01]
- 16/34 • • Browsing; Visualisation therefor (browsing or visualisation for clustering or classification G06F 16/358) [2019.01, 2025.01]
- 16/35 • • Clustering; Classification [2019.01, 2025.01]
- 16/353 • • • into predefined classes [2025.01]
- 16/355 • • • Creation or modification of classes or clusters [2025.01]
- 16/358 • • • Browsing; Visualisation therefor [2025.01]
- 16/36 • • Creation of semantic tools, e.g. ontology or thesauri [2019.01]
- 16/38 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/383 • • • using metadata automatically derived from the content [2019.01]
- 16/387 • • • using geographical or spatial information, e.g. location [2019.01]
- 16/40 • of multimedia data, e.g. slideshows comprising image and additional audio data (retrieval of still image data G06F 16/50; retrieval of audio data G06F 16/60; retrieval of video data G06F 16/70) [2019.01]

Note(s) [2019.01]

In groups G06F 16/40-G06F 16/45, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/48-G06F 16/487.

- 16/41 • • Indexing; Data structures therefor; Storage structures [2019.01]
- 16/43 • • Querying [2019.01]
- 16/432 • • • Query formulation [2019.01]
- 16/435 • • • Filtering based on additional data, e.g. user or group profiles [2019.01]
- 16/438 • • • Presentation of query results [2019.01]
- 16/44 • • Browsing; Visualisation therefor [2019.01]
- 16/45 • • Clustering; Classification [2019.01]
- 16/48 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/483 • • • using metadata automatically derived from the content [2019.01]
- 16/487 • • • using geographical or spatial information, e.g. location [2019.01]
- 16/50 • of still image data [2019.01]

Note(s) [2019.01]

In groups G06F 16/50-G06F 16/56, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/58-G06F 16/587.

- 16/51 • • Indexing; Data structures therefor; Storage structures [2019.01]
- 16/53 • • Querying [2019.01]
- 16/532 • • • Query formulation, e.g. graphical querying [2019.01]
- 16/535 • • • Filtering based on additional data, e.g. user or group profiles [2019.01]
- 16/538 • • • Presentation of query results [2019.01]
- 16/54 • • Browsing; Visualisation therefor [2019.01]
- 16/55 • • Clustering; Classification [2019.01]
- 16/56 • • having vectorial format [2019.01]
- 16/58 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/583 • • • using metadata automatically derived from the content [2019.01]
- 16/587 • • • using geographical or spatial information, e.g. location [2019.01]
- 16/60 • of audio data [2019.01]
- Note(s) [2019.01]**
- In groups G06F 16/60-G06F 16/65, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/68-G06F 16/687.
- 16/61 • • Indexing; Data structures therefor; Storage structures [2019.01]
- 16/63 • • Querying [2019.01]
- 16/632 • • • Query formulation [2019.01]
- 16/635 • • • Filtering based on additional data, e.g. user or group profiles [2019.01]
- 16/638 • • • Presentation of query results [2019.01]
- 16/64 • • Browsing; Visualisation therefor (generation of a list or set of audio data G06F 16/638) [2019.01]
- 16/65 • • Clustering; Classification [2019.01]
- 16/68 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/683 • • • using metadata automatically derived from the content [2019.01]
- 16/687 • • • using geographical or spatial information, e.g. location [2019.01]
- 16/70 • of video data [2019.01]
- Note(s) [2019.01]**
- In groups G06F 16/70-G06F 16/75, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/78-G06F 16/787.
- 16/71 • • Indexing; Data structures therefor; Storage structures [2019.01]
- 16/73 • • Querying [2019.01]
- 16/732 • • • Query formulation [2019.01]
- 16/735 • • • Filtering based on additional data, e.g. user or group profiles [2019.01]
- 16/738 • • • Presentation of query results [2019.01]
- 16/74 • • Browsing; Visualisation therefor (end-user interfaces for requesting or interacting with video content, e.g. video on demand interfaces or electronic program guides, H04N 21/472) [2019.01]
- 16/75 • • Clustering; Classification [2019.01]
- 16/78 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/783 • • • using metadata automatically derived from the content [2019.01]
- 16/787 • • • using geographical or spatial information, e.g. location [2019.01]
- 16/80 • of semi-structured data, e.g. markup language structured data such as SGML, XML or HTML (content-based retrieval of web data G06F 16/95) [2019.01]
- 16/81 • • Indexing, e.g. XML tags; Data structures therefor; Storage structures [2019.01]
- 16/83 • • Querying [2019.01]
- 16/832 • • • Query formulation [2019.01]
- 16/835 • • • Query processing [2019.01]
- 16/838 • • • Presentation of query results [2019.01]
- 16/84 • • Mapping; Conversion [2019.01]
- 16/90 • Details of database functions independent of the retrieved data types [2019.01]
- Note(s) [2019.01]**
- In groups G06F 16/90-G06F 16/906, subject matter relevant to retrieval characterised by using metadata, when it is determined to be novel and non-obvious, must also be classified in groups G06F 16/907-G06F 16/909.
- 16/901 • • Indexing; Data structures therefor; Storage structures (for retrieval from the web G06F 16/951) [2019.01]
- 16/903 • • Querying (for retrieval from the web G06F 16/953) [2019.01]
- 16/9032 • • • Query formulation [2019.01]
- 16/9035 • • • Filtering based on additional data, e.g. user or group profiles [2019.01]
- 16/9038 • • • Presentation of query results [2019.01]
- 16/904 • • Browsing; Visualisation therefor (for navigating the web G06F 16/954; browsing optimisation for the web G06F 16/957) [2019.01]
- 16/906 • • Clustering; Classification [2019.01]
- 16/907 • • Retrieval characterised by using metadata, e.g. metadata not derived from the content or metadata generated manually [2019.01]
- 16/908 • • • using metadata automatically derived from the content [2019.01]
- 16/909 • • • using geographical or spatial information, e.g. location (spatial or temporal dependent retrieval from the web G06F 16/9537) [2019.01]
- 16/93 • • Document management systems [2019.01]
- 16/95 • • Retrieval from the web [2019.01]
- 16/951 • • • Indexing; Web crawling techniques [2019.01]
- 16/953 • • • Querying, e.g. by the use of web search engines [2019.01]
- 16/9532 • • • • Query formulation [2019.01]
- 16/9535 • • • • Search customisation based on user profiles and personalisation [2019.01]
- 16/9536 • • • • Search customisation based on social or collaborative filtering [2019.01]
- 16/9537 • • • • Spatial or temporal dependent retrieval, e.g. spatiotemporal queries [2019.01]
- 16/9538 • • • • Presentation of query results [2019.01]
- 16/954 • • • • Navigation, e.g. using categorised browsing [2019.01]
- 16/955 • • • • using information identifiers, e.g. uniform resource locators [URL] [2019.01]
- 16/957 • • • • Browsing optimisation, e.g. caching or content distillation [2019.01]
- 16/958 • • • • Organisation or management of web site content, e.g. publishing, maintaining pages or automatic linking [2019.01]

- 17/00 Digital computing or data processing equipment or methods, specially adapted for specific functions** (information retrieval, database structures or file system structures therefor G06F 16/00) [6, 2006.01, 2019.01]
- 17/10 • Complex mathematical operations [6, 2006.01]
 - 17/11 • • for solving equations [6, 2006.01]
 - 17/12 • • • Simultaneous equations [6, 2006.01]
 - 17/13 • • • Differential equations (using digital differential analysers G06F 7/64) [6, 2006.01]
 - 17/14 • • Fourier, Walsh or analogous domain transformations [6, 2006.01]
 - 17/15 • • Correlation function computation [6, 2006.01]
 - 17/16 • • Matrix or vector computation [6, 2006.01]
 - 17/17 • • Function evaluation by approximation methods, e.g. interpolation or extrapolation, smoothing or least mean square method [6, 2006.01]
 - 17/18 • • for evaluating statistical data [6, 2006.01]
 - 17/40 • Data acquisition and logging (for input to computer G06F 3/00) [6, 2006.01]
- 18/00 Pattern recognition [2023.01]**
- 18/10 • Pre-processing; Data cleansing [2023.01]
 - 18/15 • • Statistical pre-processing, e.g. techniques for normalisation or restoring missing data [2023.01]
 - 18/20 • Analysing [2023.01]
 - 18/21 • • Design or setup of recognition systems or techniques; Extraction of features in feature space; Blind source separation [2023.01]
 - 18/211 • • • Selection of the most significant subset of features [2023.01]
 - 18/2111 • • • • by using evolutionary computational techniques, e.g. genetic algorithms [2023.01]
 - 18/2113 • • • • by ranking or filtering the set of features, e.g. using a measure of variance or of feature cross-correlation [2023.01]
 - 18/2115 • • • • by evaluating different subsets according to an optimisation criterion, e.g. class separability, forward selection or backward elimination [2023.01]
 - 18/213 • • • Feature extraction, e.g. by transforming the feature space; Summarisation; Mappings, e.g. subspace methods [2023.01]
 - 18/2131 • • • • based on a transform domain processing, e.g. wavelet transform [2023.01]
 - 18/2132 • • • • based on discrimination criteria, e.g. discriminant analysis [2023.01]
 - 18/2133 • • • • based on naturality criteria, e.g. with non-negative factorisation or negative correlation [2023.01]
 - 18/2134 • • • • based on separation criteria, e.g. independent component analysis [2023.01]
 - 18/2135 • • • • based on approximation criteria, e.g. principal component analysis [2023.01]
 - 18/2136 • • • • based on sparsity criteria, e.g. with an overcomplete basis [2023.01]
 - 18/2137 • • • • based on criteria of topology preservation, e.g. multidimensional scaling or self-organising maps [2023.01]
 - 18/214 • • • Generating training patterns; Bootstrap methods, e.g. bagging or boosting [2023.01]
 - 18/22 • • Matching criteria, e.g. proximity measures [2023.01]
 - 18/23 • • Clustering techniques [2023.01]
 - 18/231 • • • Hierarchical techniques, i.e. dividing or merging pattern sets so as to obtain a dendrogram [2023.01]
 - 18/232 • • • Non-hierarchical techniques [2023.01]
 - 18/2321 • • • • using statistics or function optimisation, e.g. modelling of probability density functions [2023.01]
 - 18/23211 • • • • • with adaptive number of clusters [2023.01]
 - 18/23213 • • • • • with fixed number of clusters, e.g. K-means clustering [2023.01]
 - 18/2323 • • • • based on graph theory, e.g. minimum spanning trees [MST] or graph cuts [2023.01]
 - 18/2325 • • • • using vector quantisation [2023.01]
 - 18/2337 • • • • using fuzzy logic, i.e. fuzzy clustering [2023.01]
 - 18/24 • • Classification techniques [2023.01]
 - 18/241 • • • relating to the classification model, e.g. parametric or non-parametric approaches [2023.01]
 - 18/2411 • • • • based on the proximity to a decision surface, e.g. support vector machines [2023.01]
 - 18/2413 • • • • based on distances to training or reference patterns [2023.01]
 - 18/2415 • • • • based on parametric or probabilistic models, e.g. based on likelihood ratio or false acceptance rate versus a false rejection rate [2023.01]
 - 18/243 • • • relating to the number of classes [2023.01]
 - 18/2431 • • • • Multiple classes [2023.01]
 - 18/2433 • • • • Single-class perspective, e.g. one-against-all classification; Novelty detection; Outlier detection [2023.01]
 - 18/245 • • • relating to the decision surface [2023.01]
 - 18/2451 • • • • linear, e.g. hyperplane [2023.01]
 - 18/2453 • • • • non-linear, e.g. polynomial classifier [2023.01]
 - 18/25 • • Fusion techniques [2023.01]
 - 18/26 • • Discovering frequent patterns [2023.01]
 - 18/27 • • Regression, e.g. linear or logistic regression [2023.01]
 - 18/28 • • Determining representative reference patterns, e.g. by averaging or distorting; Generating dictionaries [2023.01]
 - 18/30 • Post-processing [2023.01]
 - 18/40 • Software arrangements specially adapted for pattern recognition, e.g. user interfaces or toolboxes therefor [2023.01]
- 21/00 Security arrangements for protecting computers, components thereof, programs or data against unauthorised activity [2006.01, 2013.01]**
- 21/10 • Protecting distributed programs or content, e.g. vending or licensing of copyrighted material (protection in video systems or pay television H04N 7/16) [2013.01]
- Note(s) [2013.01]**
- In this group, the following terms or expressions are used with the meaning indicated:
- "content" means any intellectually created work whose copyright is to be safeguarded.
- 21/12 • • Protecting executable software [2013.01]
 - 21/14 • • • against software analysis or reverse engineering, e.g. by obfuscation [2013.01]
 - 21/16 • • Program or content traceability, e.g. by watermarking [2013.01]
 - 21/30 • Authentication, i.e. establishing the identity or authorisation of security principals [2013.01]
 - 21/31 • • User authentication [2013.01]

- 21/32 • • • using biometric data, e.g. fingerprints, iris scans or voiceprints **[2013.01]**
- 21/33 • • • using certificates **[2013.01]**
- 21/34 • • • involving the use of external additional devices, e.g. dongles or smart cards **[2013.01]**
- 21/35 • • • • communicating wirelessly **[2013.01]**
- 21/36 • • • • by graphic or iconic representation **[2013.01]**
- 21/40 • • • • by quorum, i.e. whereby two or more security principals are required **[2013.01]**
- 21/41 • • • where a single sign-on provides access to a plurality of computers **[2013.01]**
- 21/42 • • • using separate channels for security data **[2013.01]**
- 21/43 • • • • wireless channels **[2013.01]**
- 21/44 • • Program or device authentication **[2013.01]**
- 21/45 • • Structures or tools for the administration of authentication **[2013.01]**
- 21/46 • • • by designing passwords or checking the strength of passwords **[2013.01]**
- 21/50 • Monitoring users, programs or devices to maintain the integrity of platforms, e.g. of processors, firmware or operating systems **[2013.01]**
- 21/51 • • at application loading time, e.g. accepting, rejecting, starting or inhibiting executable software based on integrity or source reliability **[2013.01]**
- 21/52 • • during program execution, e.g. stack integrity, buffer overflow or preventing unwanted data erasure **[2013.01]**
- 21/53 • • • by executing in a restricted environment, e.g. sandbox or secure virtual machine **[2013.01]**
- 21/54 • • • by adding security routines or objects to programs **[2013.01]**
- 21/55 • • Detecting local intrusion or implementing counter-measures **[2013.01]**
- 21/56 • • • Computer malware detection or handling, e.g. anti-virus arrangements **[2013.01]**
- 21/57 • • Certifying or maintaining trusted computer platforms, e.g. secure boots or power-downs, version controls, system software checks, secure updates or assessing vulnerabilities **[2013.01]**
- 21/60 • Protecting data **[2013.01]**
- 21/62 • • Protecting access to data via a platform, e.g. using keys or access control rules **[2013.01]**
- 21/64 • • Protecting data integrity, e.g. using checksums, certificates or signatures **[2013.01]**
- 21/70 • Protecting specific internal or peripheral components, in which the protection of a component leads to protection of the entire computer **[2013.01]**
- 21/71 • • to assure secure computing or processing of information **[2013.01]**
- 21/72 • • • in cryptographic circuits **[2013.01]**
- 21/73 • • • by creating or determining hardware identification, e.g. serial numbers **[2013.01]**
- 21/74 • • • operating in dual or compartmented mode, i.e. at least one secure mode **[2013.01]**
- 21/75 • • • by inhibiting the analysis of circuitry or operation, e.g. to counteract reverse engineering **[2013.01]**
- 21/76 • • • in application-specific integrated circuits [ASIC] or field-programmable devices, e.g. field-programmable gate arrays [FPGA] or programmable logic devices [PLD] **[2013.01]**
- 21/77 • • • in smart cards **[2013.01]**
- 21/78 • • to assure secure storage of data (address-based protection against unauthorised use of memory G06F 12/14; record carriers for use with machines and with at least a part designed to carry digital markings G06K 19/00) **[2013.01]**
- 21/79 • • • in semiconductor storage media, e.g. directly-addressable memories **[2013.01]**
- 21/80 • • • in storage media based on magnetic or optical technology, e.g. disks with sectors (preventing unauthorised reproduction or copying of disc-type recordable media G11B 20/00) **[2013.01]**
- 21/81 • • by operating on the power supply, e.g. enabling or disabling power-on, sleep or resume operations **[2013.01]**
- 21/82 • • Protecting input, output or interconnection devices **[2013.01]**
- 21/83 • • • input devices, e.g. keyboards, mice or controllers thereof **[2013.01]**
- 21/84 • • • output devices, e.g. displays or monitors **[2013.01]**
- 21/85 • • • interconnection devices, e.g. bus-connected or in-line devices **[2013.01]**
- 21/86 • • Secure or tamper-resistant housings **[2013.01]**
- 21/87 • • • by means of encapsulation, e.g. for integrated circuits **[2013.01]**
- 21/88 • • Detecting or preventing theft or loss **[2013.01]**
- 30/00 Computer-aided design [CAD] [2020.01]**
- Note(s) [2020.01]**
- In this group, it is desirable to add the indexing codes of groups G06F 111/00-G06F 119/00.
- 30/10 • Geometric CAD **[2020.01]**
- 30/12 • • characterised by design entry means specially adapted for CAD, e.g. graphical user interfaces [GUI] specially adapted for CAD **[2020.01]**
- 30/13 • • Architectural design, e.g. computer-aided architectural design [CAAD] related to design of buildings, bridges, landscapes, production plants or roads **[2020.01]**
- 30/15 • • Vehicle, aircraft or watercraft design **[2020.01]**
- 30/17 • • Mechanical parametric or variational design **[2020.01]**
- 30/18 • • Network design, e.g. design based on topological or interconnect aspects of utility systems, piping, heating ventilation air conditioning [HVAC] or cabling (circuit design at the physical level G06F 30/39; network planning tools for wireless communication networks H04W 16/18) **[2020.01]**
- 30/20 • Design optimisation, verification or simulation (optimisation, verification or simulation of circuit designs G06F 30/30) **[2020.01]**
- 30/22 • • using Petri net models **[2020.01]**
- 30/23 • • using finite element methods [FEM] or finite difference methods [FDM] **[2020.01]**
- 30/25 • • using particle-based methods **[2020.01]**
- 30/27 • • using machine learning, e.g. artificial intelligence, neural networks, support vector machines [SVM] or training a model **[2020.01]**
- 30/28 • • using fluid dynamics, e.g. using Navier-Stokes equations or computational fluid dynamics [CFD] **[2020.01]**
- 30/30 • Circuit design **[2020.01]**
- 30/31 • • Design entry, e.g. editors specifically adapted for circuit design **[2020.01]**
- 30/32 • • Circuit design at the digital level (reconfigurable circuits G06F 30/34) **[2020.01]**

- 30/323 • • • Translation or migration, e.g. logic to logic, hardware description language [HDL] translation or netlist translation **[2020.01]**
- 30/327 • • • Logic synthesis; Behaviour synthesis, e.g. mapping logic, HDL to netlist, high-level language to RTL or netlist **[2020.01]**
- 30/33 • • • Design verification, e.g. functional simulation or model checking **[2020.01]**
- 30/3308 • • • using simulation **[2020.01]**
- 30/331 • • • • with hardware acceleration, e.g. by using field programmable gate array [FPGA] or emulation **[2020.01]**
- 30/3312 • • • • Timing analysis **[2020.01]**
- 30/3315 • • • • using static timing analysis [STA] **[2020.01]**
- 30/3323 • • • • using formal methods, e.g. equivalence checking or property checking **[2020.01]**
- 30/333 • • • Design for testability [DFT], e.g. scan chain or built-in self-test [BIST] **[2020.01]**
- 30/337 • • • Design optimisation **[2020.01]**
- 30/34 • • for reconfigurable circuits, e.g. field programmable gate arrays [FPGA] or programmable logic devices [PLD] **[2020.01]**
- 30/343 • • • Logical level **[2020.01]**
- 30/347 • • • Physical level, e.g. placement or routing **[2020.01]**
- 30/35 • • Delay-insensitive circuit design, e.g. asynchronous or self-timed **[2020.01]**
- 30/36 • • Circuit design at the analogue level **[2020.01]**
- 30/367 • • • Design verification, e.g. using simulation, simulation program with integrated circuit emphasis [SPICE], direct methods or relaxation methods **[2020.01]**
- 30/373 • • • Design optimisation **[2020.01]**
- 30/38 • • Circuit design at the mixed level of analogue and digital signals **[2020.01]**
- 30/39 • • Circuit design at the physical level (physical level design for reconfigurable circuits G06F 30/347) **[2020.01]**
- 30/392 • • • Floor-planning or layout, e.g. partitioning or placement **[2020.01]**
- 30/394 • • • Routing (G06F 30/396 takes precedence) **[2020.01]**
- 30/3947 • • • • global **[2020.01]**
- 30/3953 • • • • detailed **[2020.01]**
- 30/396 • • • Clock trees **[2020.01]**
- 30/398 • • • Design verification or optimisation, e.g. using design rule check [DRC], layout versus schematics [LVS] or finite element methods [FEM] (optical proximity correction [OPC] design processes G03F 1/36) **[2020.01]**
- 40/00 Handling natural language data** (speech analysis or synthesis, speech recognition G10L) **[2020.01]**
- 40/10 • Text processing (natural language analysis G06F 40/20; semantic analysis G06F 40/30; processing or translation of natural language G06F 40/40) **[2020.01]**
- 40/103 • • Formatting, i.e. changing of presentation of documents (automatic justification G06F 40/189; automatic line break hyphenation G06F 40/191) **[2020.01]**
- 40/106 • • • Display of layout of documents; Previewing **[2020.01]**
- 40/109 • • • Font handling; Temporal or kinetic typography **[2020.01]**
- 40/111 • • • Mathematical or scientific formatting; Subscripts; Superscripts **[2020.01]**
- 40/114 • • • Pagination **[2020.01]**
- 40/117 • • • Tagging; Marking up (details of markup languages G06F 40/143); Designating a block; Setting of attributes (style sheets, e.g. eXtensible Stylesheet Language Transformation [XSLT], G06F 40/154) **[2020.01]**
- 40/12 • • Use of codes for handling textual entities **[2020.01]**
- 40/123 • • • Storage facilities **[2020.01]**
- 40/126 • • • Character encoding **[2020.01]**
- 40/129 • • • • Handling non-Latin characters, e.g. kana-to-kanji conversion **[2020.01]**
- 40/131 • • • Fragmentation of text files, e.g. creating reusable text-blocks; Linking to fragments, e.g. using XInclude; Namespaces **[2020.01]**
- 40/134 • • • Hyperlinking **[2020.01]**
- 40/137 • • • Hierarchical processing, e.g. outlines **[2020.01]**
- 40/14 • • • Tree-structured documents (parsing G06F 40/205; validation G06F 40/226) **[2020.01]**
- 40/143 • • • • Markup, e.g. Standard Generalized Markup Language [SGML] or Document Type Definition [DTD] **[2020.01]**
- 40/146 • • • • Coding or compression of tree-structured data **[2020.01]**
- 40/149 • • • Adaptation of the text data for streaming purposes, e.g. Efficient XML Interchange [EXI] format **[2020.01]**
- 40/151 • • • Transformation **[2020.01]**
- 40/154 • • • • Tree transformation for tree-structured or markup documents, e.g. XSLT, XSL-FO or stylesheets **[2020.01]**
- 40/157 • • • • using dictionaries or tables **[2020.01]**
- 40/16 • • • • Automatic learning of transformation rules, e.g. from examples **[2020.01]**
- 40/163 • • • Handling of whitespace **[2020.01]**
- 40/166 • • Editing, e.g. inserting or deleting **[2020.01]**
- 40/169 • • • Annotation, e.g. comment data or footnotes **[2020.01]**
- 40/171 • • • • by use of digital ink **[2020.01]**
- 40/174 • • • Form filling; Merging **[2020.01]**
- 40/177 • • • • of tables; using ruled lines **[2020.01]**
- 40/18 • • • • of spreadsheets (form-filling G06F 40/174) **[2020.01]**
- 40/183 • • • Tabulation, i.e. one-dimensional positioning **[2020.01]**
- 40/186 • • • Templates **[2020.01]**
- 40/189 • • Automatic justification **[2020.01]**
- 40/191 • • Automatic line break hyphenation **[2020.01]**
- 40/194 • • Calculation of difference between files **[2020.01]**
- 40/197 • • Version control (for software G06F 8/71) **[2020.01]**
- 40/20 • Natural language analysis (semantic analysis of natural language G06F 40/30) **[2020.01]**
- 40/205 • • Parsing **[2020.01]**
- 40/211 • • • Syntactic parsing, e.g. based on context-free grammar [CFG] or unification grammars **[2020.01]**
- 40/216 • • • using statistical methods **[2020.01]**
- 40/221 • • • Parsing markup language streams (streaming G06F 40/149) **[2020.01]**
- 40/226 • • • Validation **[2020.01]**
- 40/232 • • Orthographic correction, e.g. spell checking or vowelisation **[2020.01]**
- 40/237 • • Lexical tools **[2020.01]**
- 40/242 • • • Dictionaries **[2020.01]**

- 40/247 • • • Thesauruses; Synonyms [2020.01]
- 40/253 • • Grammatical analysis; Style critique [2020.01]
- 40/258 • • Heading extraction; Automatic titling; Numbering [2020.01]
- 40/263 • • Language identification [2020.01]
- 40/268 • • Morphological analysis [2020.01]
- 40/274 • • Converting codes to words; Guess-ahead of partial word inputs [2020.01]
- 40/279 • • Recognition of textual entities [2020.01]
- 40/284 • • • Lexical analysis, e.g. tokenisation or collocates [2020.01]
- 40/289 • • • Phrasal analysis, e.g. finite state techniques or chunking [2020.01]
- 40/295 • • • Named entity recognition [2020.01]
- 40/30 • Semantic analysis [2020.01]
- 40/35 • • Discourse or dialogue representation [2020.01]
- 40/40 • Processing or translation of natural language (natural language analysis G06F 40/20; semantic analysis G06F 40/30) [2020.01]
- 40/42 • • Data-driven translation [2020.01]
- 40/44 • • • Statistical methods, e.g. probability models [2020.01]
- 40/45 • • • Example-based machine translation; Alignment [2020.01]
- 40/47 • • • Machine-assisted translation, e.g. using translation memory [2020.01]
- 40/49 • • • using very large corpora, e.g. the web [2020.01]
- 40/51 • • Translation evaluation [2020.01]
- 40/53 • • Processing of non-Latin text (kana-to-kanji conversion G06F 40/129; vowelisation G06F 40/232) [2020.01]
- 40/55 • • Rule-based translation [2020.01]
- 40/56 • • • Natural language generation [2020.01]
- 40/58 • • Use of machine translation, e.g. for multi-lingual retrieval, for server-side translation for client devices or for real-time translation [2020.01]

Indexing scheme associated with group G06F 30/00, relating to CAD techniques [2020.01]

111/00 Details relating to CAD techniques [2020.01]

- 111/02 • CAD in a network environment, e.g. collaborative CAD or distributed simulation [2020.01]
- 111/04 • Constraint-based CAD [2020.01]
- 111/06 • Multi-objective optimisation, e.g. Pareto optimisation using simulated annealing [SA], ant colony algorithms or genetic algorithms [GA] [2020.01]
- 111/08 • Probabilistic or stochastic CAD [2020.01]
- 111/10 • Numerical modelling [2020.01]
- 111/12 • Symbolic schematics [2020.01]
- 111/14 • related to nanotechnology [2020.01]
- 111/16 • Customisation or personalisation [2020.01]
- 111/18 • using virtual or augmented reality [2020.01]
- 111/20 • Configuration CAD, e.g. designing by assembling or positioning modules selected from libraries of predesigned modules [2020.01]

Indexing scheme associated with group G06F 30/00, relating to the application field [2020.01]

113/00 Details relating to the application field [2020.01]

- 113/02 • Data centres [2020.01]
- 113/04 • Power grid distribution networks [2020.01]
- 113/06 • Wind turbines or wind farms [2020.01]
- 113/08 • Fluids [2020.01]
- 113/10 • Additive manufacturing, e.g. 3D printing [2020.01]
- 113/12 • Cloth [2020.01]
- 113/14 • Pipes [2020.01]
- 113/16 • Cables, cable trees or wire harnesses [2020.01]
- 113/18 • Chip packaging [2020.01]
- 113/20 • Packaging, e.g. boxes or containers [2020.01]
- 113/22 • Moulding [2020.01]
- 113/24 • Sheet material [2020.01]
- 113/26 • Composites [2020.01]
- 113/28 • Fuselage, exterior or interior [2020.01]

Indexing scheme associated with group G06F 30/00, relating to the type of the circuit [2020.01]

115/00 Details relating to the type of the circuit [2020.01]

- 115/02 • System on chip [SoC] design [2020.01]
- 115/04 • Micro electro-mechanical systems [MEMS] [2020.01]
- 115/06 • Structured ASICs [2020.01]
- 115/08 • Intellectual property [IP] blocks or IP cores [2020.01]
- 115/10 • Processors [2020.01]
- 115/12 • Printed circuit boards [PCB] or multi-chip modules [MCM] [2020.01]

Indexing scheme associated with group G06F 30/00, relating to the type or aim of the circuit design [2020.01]

117/00 Details relating to the type or aim of the circuit design [2020.01]

- 117/02 • Fault tolerance, e.g. for transient fault suppression [2020.01]
- 117/04 • Clock gating [2020.01]
- 117/06 • Spare resources, e.g. for permanent fault suppression [2020.01]
- 117/08 • HW-SW co-design, e.g. HW-SW partitioning [2020.01]
- 117/10 • Buffer insertion [2020.01]
- 117/12 • Sizing, e.g. of transistors or gates [2020.01]

Indexing scheme associated with group G06F 30/00, relating to the purpose – mostly applicable to circuits – but also relevant for general CAD [2020.01]

119/00 Details relating to the type or aim of the analysis or the optimisation [2020.01]

- 119/02 • Reliability analysis or reliability optimisation; Failure analysis, e.g. worst case scenario performance, failure mode and effects analysis [FMEA] [2020.01]
- 119/04 • Ageing analysis or optimisation against ageing [2020.01]
- 119/06 • Power analysis or power optimisation [2020.01]
- 119/08 • Thermal analysis or thermal optimisation [2020.01]
- 119/10 • Noise analysis or noise optimisation [2020.01]
- 119/12 • Timing analysis or timing optimisation [2020.01]

G06F

- 119/14 • Force analysis or force optimisation, e.g. static or dynamic forces [2020.01]
- 119/16 • Equivalence checking [2020.01]
- 119/18 • Manufacturability analysis or optimisation for manufacturability [2020.01]
- 119/20 • Design reuse, reusability analysis or reusability optimisation [2020.01]
- 119/22 • Yield analysis or yield optimisation [2020.01]

Indexing scheme associated with group G06F 18/00, relating to pattern recognition [2023.01]

123/00 Data types [2023.01]

- 123/02 • in the time domain, e.g. time-series data [2023.01]