

SECTION G — PHYSICS

G06 COMPUTING; CALCULATING; COUNTING

G06F ELECTRIC DIGITAL DATA PROCESSING (computers in which a part of the computation is effected hydraulically or pneumatically G06D, optically G06E; 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, 15/00-19/00
INPUT, OUTPUT; INTERCONNECTIONS BETWEEN FUNCTIONAL ELEMENTS.....	3/00, 13/00
ADDRESSING OR ALLOCATION.....	12/00
CONVERSION; PROGRAMME CONTROL; ERROR DETECTION, MONITORING.....	5/00, 9/00, 11/00
DETAILS.....	1/00
SECURITY ARRANGEMENTS.....	21/00

1/00 Details not covered by groups G06F 3/00-G06F 13/00 and G06F 21/00 (architectures of general purpose stored programme 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 (restoration from data faults G06F 11/00) **[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; involving the processing of data-words G06F 11/00) **[5, 2006.01]**
- 1/32 • • Means for saving power **[5, 2006.01]**

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 [GUIs] [2006.01, 2013.01]

Note(s) [2013.01]

1. 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.
 2. In this group, multi-aspect classification is applied, so that subject matter characterised by aspects covered by more than one of its subgroups, which is considered to represent information of interest for search, should be classified in each of those subgroups.
- 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]
 - 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 or an image, setting a parameter value or selecting a range [2013.01]
- 3/0485 • • • • Scrolling or panning [2013.01]
- 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]
- 3/0489 • • • • using dedicated keyboard keys or combinations thereof [2013.01]
- 3/05 • Digital input using the sampling of an analogue quantity at regular intervals of time (sample-and-hold arrangements G11C 27/02) [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 (arrangements for producing a permanent visual presentation of the output data using printers G06K 15/02) [1, 2006.01]
- 3/13 • Digital output to plotter (arrangements for producing a permanent visual presentation of the output data using plotters G06K 15/22) [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 (conversion of speech into digital information or vice versa 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, 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 Sweeny-Robertson-Tocher (SRT) algorithm [2006.01]
- 7/544 • • • for evaluating functions by calculation (with a look-up table G06F 1/02) [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, LIFO buffers; Overflow or underflow handling therefor [2006.01]
- 9/00 **Arrangements for programme control, e.g. control unit** (programme control for peripheral devices G06F 13/10) [1, 4, 2006.01]
- 9/02 • using wired connections, e.g. plugboard [1, 2006.01]
- 9/04 • using record carriers containing only programme instructions (G06F 9/06 takes precedence) [1, 2006.01]
- 9/06 • using stored programme, i.e. using internal store of processing equipment to receive and retain programme [1, 2006.01]
- 9/22 • • Micro-control or micro-programme arrangements [3, 2006.01]
- 9/24 • • • Loading of the micro-programme [3, 2006.01]
- 9/26 • • • Address formation of the next micro-instruction (G06F 9/28 takes precedence) [3, 2006.01]
- 9/28 • • • Enhancement of operational speed, e.g. by using several micro-control devices operating in parallel [3, 2006.01]
- 9/30 • • Arrangements for executing machine- instructions, e.g. instruction decode (for executing micro-instructions G06F 9/22; for executing subprogrammes G06F 9/40) [3, 2006.01]
- 9/302 • • • Controlling the executing of arithmetic operations [5, 2006.01]
- 9/305 • • • Controlling the executing of logical operations [5, 2006.01]
- 9/308 • • • Controlling single bit operations (G06F 9/305 takes precedence) [5, 2006.01]
- 9/312 • • • Controlling loading, storing or clearing operations [5, 2006.01]
- 9/315 • • • Controlling moving, shifting or rotation operations [5, 2006.01]
- 9/318 • • • with operation extension or modification [5, 2006.01]
- 9/32 • • • Address formation of the next instruction, e.g. incrementing the instruction counter, jump (G06F 9/38 takes precedence; subprogramme jump G06F 9/42) [3, 2006.01]
- 9/34 • • • Addressing or accessing the instruction operand or the result [3, 5, 2006.01]
- 9/345 • • • • of multiple operands or results [5, 2006.01]
- 9/35 • • • • Indirect addressing [5, 2006.01]
- 9/355 • • • • Indexed addressing [5, 2006.01]
- 9/38 • • • Concurrent instruction execution, e.g. pipeline, look ahead [3, 2006.01]
- 9/40 • • Arrangements for executing subprogrammes, i.e. combinations of several instructions [3, 2006.01]
- 9/42 • • • Formation of subprogramme-jump address or of return address [3, 2006.01]
- 9/44 • • Arrangements for executing specific programmes [3, 2006.01]
- 9/445 • • • Programme loading or initiating [5, 2006.01]
- 9/45 • • • Compilation or interpretation of high level programme languages [5, 2006.01]
- 9/455 • • • Emulation; Software simulation [5, 2006.01]
- 9/46 • • Multiprogramming arrangements [3, 2006.01]
- 9/48 • • • Programme initiating; Programme 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 • • • Programme synchronisation; Mutual exclusion, e.g. by means of semaphores [7, 2006.01]
- 9/54 • • • Interprogramme communication [7, 2006.01]
- 11/00 **Error detection; Error correction; Monitoring** (methods or arrangements for verifying the correctness of marking on a record carrier G06K 5/00; in information storage based on relative movement between record carrier and transducer G11B, e.g. G11B 20/18; 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 (testing of digital circuits, e.g. of separate computer components, G01R 31/317) [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 • Preventing errors by testing or debugging of software [7, 2006.01]
- 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 (programme 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 • Programme 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 programme, 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 programme unit and a register, e.g. for a simultaneous processing of several programmes [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]
- 15/18 • in which a programme is changed according to experience gained by the computer itself during a complete run; Learning machines (adaptive control systems G05B 13/00) [1, 2006.01]
- 15/76 • Architectures of general purpose stored programme computers (with programme plugboard G06F 15/08; multicomputers G06F 15/16; general purpose image data processing G06T 1/00) [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]
- 17/00 Digital computing or data processing equipment or methods, specially adapted for specific functions [6, 2006.01]**
- 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]
- 17/20 • Handling natural language data (speech analysis or synthesis G10L) [6, 2006.01]
- 17/21 • • Text processing (G06F 17/27, G06F 17/28 take precedence) [6, 2006.01]
- 17/22 • • • Manipulating or registering by use of codes, e.g. in sequence of text characters [6, 2006.01]
- 17/24 • • • Editing, e.g. insert/delete [6, 2006.01]
- 17/25 • • • Automatic justification [6, 2006.01]
- 17/26 • • • Automatic hyphenation [6, 2006.01]
- 17/27 • • Automatic analysis, e.g. parsing, orthograph correction [6, 2006.01]
- 17/28 • • Processing or translating of natural language (G06F 17/27 takes precedence) [6, 2006.01]
- 17/30 • Information retrieval; Database structures therefor [6, 2006.01]
- 17/40 • Data acquisition and logging (for input to computer G06F 3/00) [6, 2006.01]
- 17/50 • Computer-aided design (for the design of test circuits for static stores G11C 29/54) [6, 2006.01]
- 19/00 Digital computing or data processing equipment or methods, specially adapted for specific applications** (G06F 17/00 takes precedence; data processing systems or methods specially adapted for administrative, commercial, financial, managerial, supervisory or forecasting purposes G06Q) [6, 2006.01, 2011.01]
- Note(s) [6]**
- This group covers:
- special constructions of computers to permit or facilitate use in specific applications;
 - non-structural adaptations of computers to a specific application, e.g. computing methods.
- 19/10 • Bioinformatics, i.e. methods or systems for genetic or protein-related data processing in computational molecular biology (in silico methods of screening virtual chemical libraries C40B 30/02; in silico or mathematical methods of creating virtual chemical libraries C40B 50/02) [2011.01]
- Note(s) [2011.01]**
1. This group also covers bioinformatics methods or systems where digital data processing is inherent or implicit, but not explicitly mentioned.
2. In this group, the following term is used with the meaning indicated:
- "systems" includes apparatus.
3. 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.
- 19/12 • • for modelling or simulation in systems biology, e.g. probabilistic or dynamic models, gene-regulatory networks, protein interaction networks or metabolic networks [2011.01]
- 19/14 • • for phylogeny or evolution, e.g. evolutionarily conserved regions determination or phylogenetic tree construction [2011.01]
- 19/16 • • for molecular structure, e.g. structure alignment, structural or functional relations, protein folding, domain topologies, drug targeting using structure data, involving two-dimensional or three-dimensional structures [2011.01]
- 19/18 • • for functional genomics or proteomics, e.g. genotype-phenotype associations, linkage disequilibrium, population genetics, binding site identification, mutagenesis, genotyping or genome annotation, protein-protein interactions or protein-nucleic acid interactions [2011.01]
- 19/20 • • for hybridisation or gene expression, e.g. microarrays, sequencing by hybridisation, normalisation, profiling, noise correction models, expression ratio estimation, probe design or probe optimisation [2011.01]
- 19/22 • • for sequence comparison involving nucleotides or amino acids, e.g. homology search, motif or SNP [Single-Nucleotide Polymorphism] discovery or sequence alignment [2011.01]
- 19/24 • • for machine learning, data mining or biostatistics, e.g. pattern finding, knowledge discovery, rule extraction, correlation, clustering or classification [2011.01]
- 19/26 • • for data visualisation, e.g. graphics generation, display of maps or networks or other visual representations [2011.01]
- 19/28 • • for programming tools or database systems, e.g. ontologies, heterogeneous data integration, data warehousing or computing architectures [2011.01]
- 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 (digital watermarking on images H04N 1/32) [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 [ASICs] or field-programmable devices, e.g. field-programmable gate arrays [FPGAs] or programmable logic devices [PLDs] **[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 disk-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]**