Registers:- bits for a processor. it sets and

REGISTERS:-Registers arecircuits typically composed of flipflops, often with many characteristics similar to memory, such as: Theability to read or write multiple bits ata time, andUsingan address toselect a particular register in a manner similar to a memoryaddress. Theirdistinguishing characteristic, Theyalso have special hardware-related functions beyond those ofordinary memory. Alsomay only use to store data. Thereare two kind of registers:- GeneralPurpose Registersii)Special Purpose RegisterGENERALPURPOSE REGISTERSTheseregisters are user accessible and each register can hold 8 bit ofdata. Whenwe want to load/store data of more than 8 bit then these can be usedas pair.

CommonGPR are: – B, C, D, E, H and L. Pairing:- B-C, D-E, H-L. SPECIALPURPOSE REGISTERSTheseregisters have some specific task to perform. In it each register isassign to unit or grouped such that to make a functional unit. Thesecategory is further divided into two categories:- UserAccessibleii) User InaccessibleUseraccessible special purpose registerAsname suggests, they could be used by the user for storing data or toperform the various tasks. The user accessible SPR are Accumulatorand Status Register (Flag Register).
Accumulator isa register inwhich intermediate arithmeticand logic resultsare stored.

Ifwe won’t have accumulator then we have to store the resultsomewhere else or in main memory. Accessto main memory is slower than access to a register like theaccumulator because the technology used for the large main memory isslower (but cheaper) than that used for a register. StatusRegister isa collection of status flag bits fora processor. Itsets and resets itself

according to the result obtained inaccumulator after arithmetic or logical operations for the furtherprocessing in the processor. Thereare common 4 types of flags:-Zeroflag: – If it’s high or set to 1 then it indicates that the resultof an arithmetic or logical operation (or, sometimes, a load) waszero. Otherwise it will set to 0. Carryflag: – Enables numbers larger than a single word to beadded/subtracted by carrying a binary digit from a less significantword to the leastsignificant bit ofa more significant word as needed. Ifit is high (i.

e. 1), it means the bits of result is more thancapacity of accumulator. Itis also used to extend bitshifts androtates in a similar manner on many processors (sometimes done via adedicated X flag). Signflag (or Negative flag): – Indicates that the result of amathematical operation is negative. In some processors, the N and Sflags are distinct with different meanings and usage. Oneindicates whether the last result was negative whereas the otherindicates whether a subtraction or addition has taken place. Overflowflag: – Indicates that the signed result of an operation is toolarge to fit in the register width using two’scomplement representation. Otherkind of flags: – Halfcarry flag/ Auxiliary flag: – Indicates that a bit carry wasproduced between the nibbles (typicallybetween the 4-bit halves of a byte operand) as a result of the lastarithmetic operation.

Sucha flag is generally useful for implementing BCD arithmeticoperations on binary hardware. Parityflag: – Indicates whether the number of set bits of the last resultis odd or even. It its high (i. e. 1), it means that there is evenparity and vice versa.

Interruptflag: – On some processors, this bit indicates whether interruptsare enabled or masked. NOTE:- All these flags are in different-different combinations indifferent microprocessor. STACKS:- A stackregister isa computer central processorregister whosepurpose is to keep track of a callstack. Onan accumulator-basedarchitecture machine, this may be a dedicated register such as SP (Stack- pointer). Instack pointer the last program's request is stored. It stores thedata on the top of it.

Whenever a new request appears it pushes downthe previous data and store the new one the top of it. OTHERREGISTERS: – Dataregister: – hold the data read from the memory. Addressregister: – hold the memory address of the data that would be usedfurther in any operation.

Instructionregister: – hold the current instruction i. e. to be proceed.

Inputand Output register: – it will hold the character received from theinput device and output resister hold the character that has to besent to an Output device. SIregister: – it will hold index of the source memory. ARITHMETICLOGICAL UNIT (ALU):-An arithmeticlogic unit (ALU)is a combinational digitalelectronic circuit thatperforms arithmetic and bitwiseoperations on integer binarynumbers. ALUis a fundamental building block of many types of computing circuits, including the centralprocessing unit (CPU)of computers, FPUs, and graphicsprocessing units (GPUs). A single CPU, FPU or GPU may contain multiple ALUs.

Theinputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed; the ALU'soutput is the result

of the performed operation. Inmany designs, the ALU also has status inputs or outputs, or both, which convey information about a previous operation or the currentoperation, respectively, between the ALU and external statusregisters. Operationsperformed by ALU: -Add: A and B are summed and the sum appears at Y and carry-out. Addwith carry: A, B and carry-in are summed and the sum appears at Y andcarry-out. Subtract: B is subtracted from A (or vice versa) and the difference appearsat Y and carry-out. For this function, carry-out is effectively a“ borrow” indicator. This operation may also be used tocompare the magnitudes of A and B; in such cases the Y output maybe ignored by the processor, which is only interested in the statusbits (particularly zero and negative) that result from theoperation. Subtractwith borrow: B is subtracted from A (or vice versa) with borrow (carry-in) andthe difference appears at Y and carry-out (borrow out).

Two’scomplement (negate): A (or B) is subtracted from zero and the difference appears at Y. Increment:– A (or B) is increased by one and the resulting value appears atY. Decrement:–A (or B) is decreased by one and the resulting value appears at Y. Passthrough:– all bits of A (or B) appear unmodified at Y.

This operation istypically used to determine the parity of the operand or whether itis zero or negative, or to load the operand into a processorregister. Bitwiselogical operationsAND: the bitwise AND of A and B appears at Y. OR: the bitwise OR of A and B appears at Y. Exclusive-OR: the bitwise XOR of A and B appears at Y.

Ones’complement: all bits of A (or B) are inverted and appear at Y. BitShift operations:-ALUshift operations cause operand A (or B) to shift left or right(depending on the op-code) and the shifted operand appears at Y. Simple ALUs typically can shift the operand by only one bit position, whereas more complex ALUs employ barrelshifters thatallow them to shift the operand by an arbitrary number of bits in oneoperations. Arithmeticshift: the operand is treated as a two’scomplement integer, meaning that the most significant bit is a “ sign” bit andis preserved. Logicalshift: a logic zero is shifted into the operand.

This is used to shiftunsigned integers. Rotate: the operand is treated as a circular buffer of bits so its least andmost significant bits are effectively adjacent. Rotatethrough carry: the carry bit and operand are collectively treated as a circularbuffer of bits. CONTROLUNITThe controlunit (CU)is a component of a computer’s centralprocessing unit (CPU)that directs the operation of the processor. It tells the computer’smemory, arithmetic/logic unit and input and output devices how torespond to a program’s instructions. Itdirects the operation of the other units by providing timing andcontrol signals. Most computer resources are managed by the CU. Itdirects the flow of data between the CPU and the other devices.

FUNCTIONS:-TheControl Unit (CU) is digital circuitry contained within theprocessor that coordinates the sequence of data movements into, outof, and between a processor’s many sub-units. Theresult of these routed data movements through various digitalcircuits (sub-units) within the processor produces the manipulateddata expected by a software instruction (loaded earlier, likely frommemory). Itcontrols (conducts) data flow inside the

processor and additionallyprovides several external control signals to the rest of thecomputer to further direct data and instructions to/from processorexternal destinations (i. e. memory). INSTRUCTIONTYPES: - Instructionis a command or statement that is given to computer (processor) toperform a specific task or operation. Andset or group of instruction in a correct manner is program.

Formatof Instruction:- An instruction could be of different size and will consist offollowing parts:-Op-code:– It is the special type of code understood by processor to operateitself and given by the user but already defined to processor. Address:– Address will designate a memory address or a processor registerfor the operand. Mode:– A mode is the field that is responsible for the calculation ofeffective address. Types:- ZeroAddress Instruction: -An instruction which do not have or do not require an address fieldare known as “ zero address instruction”.

So such instruction needthe implied address i. e. within the op-code. Through the impliedaddress the processor can specify the value may be using a stackpointer so that the effective address is automatically incrementedand decremented. Ex: – PUSH A, POP. OneAddress Instruction:- It uses the implied accumulator for all of its operation.

Ex: – ADDA TwoAddress Instruction: -These types of instruction are very common in this each address fieldspecify either a processor register or a memory operand. Ex: – ADDR1, A. ThreeAddress Instruction:- these instruction use each address field to specify a register or amemory location. Ex: – ADD R1, A, B. R1? M A + M B{MA- memory address of A and MB-

memory address ofB}INSTRUCTIONSETS:- Instruction sets are divided in the groups depending upon thefunction they perform. The various kind are described below: – 1. DataTransfer Instruction:- They are instructions that are used to transfer the data from onememory/register location to another without changing the content ofdata.

Mostcommonly used data transfer instructions are: – LOADSTOREMOVEEXCHANGEINPUTOUTPUTPUSHPOPSomeof the system may change the mnemonic code to specify immediateaddressing mode as in this mode usually the last alphabet of code is” I”.