**ASSIGNMENT- 3**

**CIS-580 INTRO TO COMPUTER ARCHEITECTURE SECTION 50**

**ANIL PAVULURU**

**1 ANS)**

1. Minimum hamming distance is 2
2. Minimum hamming distance is 2
3. Minimum hamming distance is 2

**2 ANS)**

The divider is 1001

And the data is 1011

As the length of the divider is 4 we have to add 4-1=3 zeros to the end of the data word.

1 0 1

1001 1 0 1 1 0 0 0

1 0 0 1

0 0 1 0 0 0

1 0 0 1

0 0 0 1 0

So, the CRC, 010 is added to the message1011. The transmitted message is 1011010.

Q3)

Find out the Hamming code for the following Data words (Show each step, 6 points):

a) 1001000

b) 1100001

c) 1101101

3 Ans)

a)

* Data is =1001000(m1=1,m2=0,m3=0,m4=1,m5=0,m6=0,m7=0)
* 2r ≥ m + r + 1
* Here m is for message bit is which is 7
* 2r ≥ m + r + 8
* So for satisfying this r must be more than 4 or more then 3
* So here it need P1,P2,P3,P4 at least as so put data from right to left with P1, P2, P3, P4 on 2 as it is and place

1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | P4 | 1 | 0 | 0 | P3 | 1 | P2 | P1 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

2) Consider that it is even parity

P1= (1, 3, 5, 7, 9, 11)

= ((P1),10100) so if it is even parity bit then P1 must be 0 to satisfied even parity bit

P1=0

P2= (2, 3, 6, 7, 10, 11)

= ((P2)10100) so here also combination of bits is 2 so already even bit so p2 must be 0

P2=0

P3= (4, 5, 6, 7)

= ((P3),001) so here also combination of bits is 1 so it is not even parity bit so p3 must be 1 for even parity bit continue

P3=1

P4= (8, 9, 10, 11)

= ((P4),000) so here also combination of bits is 0 which is already even so no need to add so

P4=0

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

-----------------------------------------------------------------------------------------------------------------------------------------

b)

* Data is =1100001(m1=1,m2=1,m3=0,m4=0,m5=0,m6=0,m7=1)
* 2r ≥ m + r + 1
* Here m is for message bit which is 7
* 2r ≥ m + r + 8
* So for satisfying this r must be more than 4 or more then 3
* So here it need P1,P2,P3,P4 at least as so put data from right to left with P1, P2, P3, P4 on 2 as it is and place

1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | P4 | 0 | 0 | 0 | P3 | 1 | P2 | P1 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

2) Consider that it is even parity

P1= (1, 3, 5, 7, 9, 11)

= ((P1),1111) so if it is even parity bit then P1 must be even and here 4 which is already even so no need to add 1

P1=0

P2= (2, 3, 6, 7, 10, 11)

= ((P2)10101) so here also combination of bits is 3 so already not even bit so p2 must be add 1 in P2

P2=1

P3= (4, 5, 6, 7)

= ((P3), 101) so here also combination of bits is 2 so it is not even parity bit so no need to add

P3=0

P4= (8, 9, 10, 11)

= ((P4), 101) so here also combination of bits is 2 which is even so no need to add 1 so

P4=0

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

------------------------------------------------------------------------------------------------------------------------------------------

c)

* Data is =1101101(m1=1,m2=1,m3=0,m4=1,m5=1,m6=0,m7=1)
* 2r ≥ m + r + 1
* Here m is for message bit which is 7
* 2r ≥ m + r + 8
* So for satisfying this r must be more than 4 or more then 3
* So here it need P1,P2,P3,P4 at least as so put data from right to left with P1, P2, P3, P4 on 2 as it is and place

1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | P4 | 1 | 0 | 1 | P3 | 1 | P2 | P1 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

2) Consider that it is even parity

P1= (1, 3, 5, 7, 9, 11)

= ((P1),1111) so if it is even parity bit then P1 must be even and here 4 which is already even so no need to add 1

P1=0

P2= (2, 3, 6, 7, 10, 11)

= ((P2)10101) so here also combination of bits is 3 so already not even bit so p2 must be add 1 in P2

P2=1

P3= (4, 5, 6, 7)

= ((P3), 101) so here also combination of bits is 2 so it is not even parity bit so no need to add

P3=0

P4= (8, 9, 10, 11)

= ((P4), 101) so here also combination of bits is 2 which is even so no need to add 1 so

P4=0

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**4 ANS)**

Output for the microarchitecture level:

* YES, in micro architecture level the execution of data path is managed by micro – program. So, the compiler is able to generate output for the micro architecture level instead of ISA level.

|  |  |  |
| --- | --- | --- |
| S.no | Pros | Cons |
| 1 | Due to elimination of interpretation program might be faster | It generates huge amount of micro - program |
| 2 | Independent of ISA level Program | It makes programs complex |
| 3 | Basically, the microprograms are placed in ROM | To make microprograms changeable, it will have to place in RAM, which makes it slow. |

**5 ANS)**

Hardware are the objects, machinery and the physical parts which the computers and the accessories are made up of while the computer software is a collection of instruction in the computer which are also known as computer programs. Software are different from hardware is physical and software is not. But some software and hardware are interdependent and both of them have to work in collaboration to make something work. Thus, hardware is useless without software and it cannot work without hardware. both of them in collaboration actually make the system work. So that this is the equivalence between them.

**6ANS)**

Key points

* Each bit in a cache is made up of transistors if we want to increase the size of the cache, we have to increase the number of transistors.
* By increasing the number of transistors, it makes the cache physical size bigger.
* Bigger the size, bigger will be the distance from the ALU.
* As the distance from the processor increase the speed of the cache decreases.
* Thus, two or more caches are required.
  + L1(primary cache) – smaller in size, nearer to the CPU, low in capacity but faster in speed.
  + L2(Secondary cache) – bigger in size, near to the CPU, having a high capacity but slower.

**7 ANS)**

Many computer architects spend a lot of time making their pipelines deeper: Since pipelines deeper gives more advantages

so the many computer architects spend a lot of time making their pipelines deeper and achieve the below advantages.

1. pipeline is made "deeper" then the step can be implemented with simpler circuitry.
2. pipeline is made "deeper" so that it can achieve processor clock run faster.
3. It leads to penalty of a misprediction increases and more number of instructions are flushed.
4. pipeline is made "deeper" effects the performance where it found to be performance gains.
5. pipeline is made "deeper" leads to speed up the execution.

**8 ANS)**

Here page size is 4096. So, page offset is 12bits.

Physical address is (frame number\* page size + page offset)

1. Virtual address 50(0k-4k). Contain page frame number 2. Page offset is 50.

Physical address is 8242.

1. Virtual address is 4400(4k-8k). contains page frame number 1. Page offset is (4400-4096) +1(including the given address) = 305.

Physical address is 4401.

1. Virtual address is 8550 (8k-12k). contain page frame number 6. Page offset is (8550-8192) +1 = 359

Physical address is 24935.

**9 ANS)**

First fit:

a.30 MB

b.10 MB

c.18 MB (assuming a and b are being included for this question)

Best fit:

a.12 MB

b.10 MB

c.9 MB

Worst fit:

a.30 MB

b.18 MB

c.15 MB

Next fit:

a.30 MB

b.18 MB

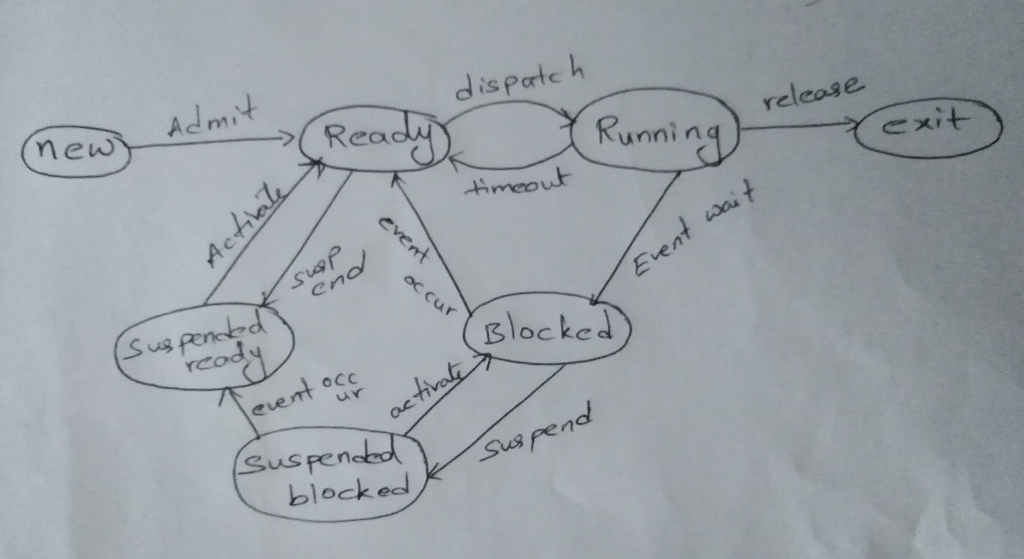
c.9 MB

**10 ANS)**

Given that

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Page | Loaded | Last ref | R | M |
| 0 | 126 | 280 | 1 | 0 |
| 1 | 230 | 265 | 0 | 1 |
| 2 | 140 | 270 | 0 | 0 |
| 3 | 110 | 285 | 1 | 1 |

1. According to second Chance, it will replace page 0 because it is loaded at 126
2. According to LRU, it will replace page 1 because it is reference at 265 (least recent)
3. According to NRU, it will replace page 2 because R = 0 and M = 0
4. According to FIFO, it will replace page 3 because it is loaded first at 110.

**11 ANS)**

1. **new**: In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by OS to create the process.
2. **READY**: Ready to run. After the creation of a process, the process enters the ready state i.e. the process is loaded into the main memory. The process here is ready to run and is waiting to get the CPU time for its execution. Processes that are ready for execution by the CPU are maintained in a queue for ready processes.
3. **RUNNING**: The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU cores.
4. **Blocked or wait**: Whenever the process requests access to I/O or needs input from the user or needs access to a critical region(the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to the ready state.
5. **terminated or completed**: Process is killed as well as PCB is deleted. Process Control Block(PCB) is a data structure in the operating system kernel containing the information needed to manage the scheduling of a particular process.
6. **suspend ready**: Process that was initially in the ready state but were swapped out of main memory and placed onto external storage by scheduler are said to be in suspend ready state. The process will transition back to ready state whenever the process is again brought onto the main memory. This generally happens when there is lack of storage in man memory.
7. **suspend blocked:** Similar to suspend ready. A process can be suspended for a number of reasons; the most signicant of which arises from the process being swapped out of memory by the memory management system in order to free memory for other processes. Other common reasons for a process being suspended are when one suspends execution while debugging a program, or when the system is monitoring processes.

When work is finished it may go to suspend ready.

A process in the SUSPEND BLOCKED\* state is moved to the SUSPEND READY state when the event for which it has been waiting occurs.

1. **long term scheduler:** it is also called jobs scheduler. A long term scheduler determines which programs are admitted to the system for processing .it selects process from the queue and loads them into memory from execution. Process loads into the memory for CPU scheduling. Primary aim of the Job Scheduler is to maintain a good degree of Multiprogramming. An optimal degree of Multiprogramming means the average rate of process creation is equal to the average departure rate of processes from the execution memory.
2. **short term scheduler**: It is also called as CPU scheduler. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects a process among the processes that are ready to execute and allocates CPU to one of them. Short-term schedulers, also known as dispatchers, make the decision of which process to execute next. Short-term schedulers are faster than long-term schedulers.

**medium term scheduler:** Medium-term scheduling is a part of swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

A running process may become suspended if it makes an I/O request. A suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other processes, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix. as already explained above medium term scheduler is responsible for swapping long term scheduler controls the degree of multi programming

1. new processes enter into the new state

when a process is ready to execute enters into the ready state

if that process requires any i/o operation to be done then that will be moved to the blocked state, if there is lack of storage in main memory that process will be moved to suspend blocked state

when the i/o is completed then that process again moved to ready state when i/o operation of the process in suspend blocked completes it will be moved to suspend ready or blocked state (according the situation will happen)

processes will be moved from suspend ready to ready when that is moved again to main memory from secondary memory when the process completes its execution then it exits