|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| University Name | | JNTU-H | | | | Semester | 5 | |
| Subject Name | | Computer Organisation and Operating Systems | | | | Topic No | 24 | |
| Subject Code | | 027S5\_A50510\_U1T24\_SB\_ENG | | | | Unit/Module No | 1 | |
| Topic Name | | Logic Micro Operations - Some Applications | | | | SB Started Date | 17/04/17 | |
| Short Name | | Logic Micro-Operations - Part B | | | | SB Completed Date | 19/04/17 | |
| ID Name  **I**  **E** | | Lekha Prabhu | | | Creative Lead (CL)Name | | V.Jayanthinathan | |
| CL Reviewed Date | | 21/04/17 | |
| PM Name | | Kiruba/Srisai | | | PM Review Started Date | | 21/04/17 | |
| PM Review completed | | 21/04/17 | |
| GD Name | | Sam Team | | | GD Start Date | | 25/04/17 | |
| GD Completed Date | | 28/04/17 | |
| Internal SME Name | | Malla Sudhakar | | | Review Started Date | | 21/04/17 | |
| Review Completed | | 21/04/17 | |
| LV Name  **I**  **E** | | Sreedevi Rajagopal | | | Date Sent to LV | | 19/04/17 | |
| Date Received from LV | | 21/04/17 | |
| External SME Name | | Malla Sudhakar | | | Date Sent to External SME | | 21/04/17 | |
| SME External QC Name (Pre SBQC) /Reviewed Date | | Malla Sudhakar | | 21/04/17 | Received from Ex-SME | | 21/04/17 | |
| No of Images | 0 | No of Videos/Animations | 8 | | No of activity | 1 | No of Slides | 11 |

After QC and GD verified the SB:

|  |  |
| --- | --- |
| SVN Path |  |
| MTFS Path- Audio and Video |  |

*Note: The Check box for “I = Internal and E = External” if checkbox is not enabled, then update I/E next to the names in a bracket.*

**Topic Description**

|  |
| --- |
| ***Topic Name***- Logic Micro Operations - Some Applications |
| Explains about the applications of logic micro-operations |

|  |  |
| --- | --- |
| **Hints**: | Any instruction or hints to voice team/GD/others |
| ***(To GDQC & QC) –*** Images & video links used in the entire SB are only for just reference; no need to draw or animate, as it is; due to font size & space concern, labels can be displayed wherever possible.  ***(To GD)* -** Use fade-in & fade-out when we go to the next scene in the video, if necessary.  ***(To All)* -** To magnify the image, just double-click on it. | |

Learning Objective

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Learning Objective | | Screen No | Heading /Frame Title | | Video/Animation/  Image No | Integration-Description |
| 1 | Logic Micro-Operations - Part B | |  | Template Introduction/ Content Template |
| Audio | | | | TOS | | GD-Description |
| V1\_1 | Logic Micro-Operations - Part B  At the end of this lesson, you will be able to:  Define logic micro-operations. | | | At the end of this lesson, you will be able to:   * Define logic micro-operations | | - |
| V1\_2 | Explain the applications of logic micro-operations. | | | * Explain the applications of logic micro-operations | |

Introduction and Procedure Learning

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Introduction / Application | | Screen No | Heading /Frame Title | | Video/Animation/Image No | Integration-Description |
| 2 | Introduction to Logic Micro-Operations | | Video2 | Video Template |
| Bulletin Label | | | | TOS | | GD-Description |
|  | | | |
| V2\_1 | Before heading to the topic, let’s discuss about the basics of logic micro-operations.  A logic micro-operation is the binary operation for strings of bits stored in registers and considers each bit of the register as a binary variable. | | | **Logic Micro-Operation** is the binary operation for strings of bits stored in registers and considers each bit as a binary variable. | | Animate:  Show the image and when the VO reads <<string of bits>> highlight yellow boxes one by one.  Show the TOS in definition strip |
| V2\_2 | There are sixteen logic micro-operations as listed in the table. | | | **Logic Micro-Operations** | | Show the table. |
| V2\_3 | In the first column of the table, the sixteen Boolean functions of two variables ‘X’ and ‘Y’ are expressed in algebraic form. | | | Highlight the 1st column one by one. And add callouts “X” and “Y” as per VO. |
| V2\_4 | The logic micro-operations are derived from these Boolean functions by replacing variables ‘X’ and ‘Y’ by the binary content of registers ‘A’ and ‘B’, respectively. | | | Highlight the 2nd column one by one. And add callouts “X”, “Y”, “A” and “B” as per VO. |
| V2\_5 | You know that among the sixteen logic micro-operations, most commonly used operations are AND, OR, Exclusive-OR and complement. | | | AND 🡪  OR 🡪 V  XOR 🡪  Complement of register A 🡪 | | Display the TOS and highlight as per VO. |
| V2\_6 | The examples discussed in this lesson, show how the bits of one register are manipulated by logic micro-operations as a function of the bits of another register. | | | Register A 🡪 1010  Register B 🡪 1100 | | When the VO reads <<one register>> type the 1st line in the TOS.  When the VO reads<< logic>> add callouts “AND”, “OR”, “NOT”.  When the VO reads <<another register>> type the 2nd line in the TOS. |
| V2\_7 | These logical operations are rarely used in scientific computations, but they are very useful for bit manipulation of binary data and for taking logical decisions. | | |  | | Animate the image as the bits are getting changed. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Screen No | | Heading / Title | | Video/Animation/Image No | | Integration -Description |
| 3 | | Selective-set Operation | | Video3 | | Video Template |
| Bulletin Label | | | TOS | | GD-Description | |
|  | | |
| V3\_1 | Let’s discuss the selective-set operation performed in register ‘A’ with a value one-zero-one-zero and register ‘B’ with a value one-one-zero-zero. | | Register A 🡪 1010  Register B 🡪 1100 | | Animate the TOS as getting typed as per VO. | |
| V3\_2 | In this operation, the bits in register ‘A’ are set to one where there are corresponding one’s in register ‘B’. It will not affect the bit positions that have zeros in register ‘B’. | | Bits in register A are set to 1 where there are corresponding 1’s in register B. | | Show the TOS in note strip. | |
| V3\_3 | The numerical example of this operation is shown here. | |  | | Show the image. | |
| V3\_4 | In this example, the two leftmost bits of register ‘B’ are ones, so the corresponding bits of ‘A’ are set to one. | |  | | Show the image and when the VO reads <<the two left>> highlight 1’s in green box.  When the VO reads << the corresponding bits>> highlight 1’s in orange box. | |
| V3\_5 | Here, one of these two bits was already set to one, so only the other bit has been changed from zero to one. | |  | | Display the image and when the VO reads <<One of these>> highlight 1 in green box.  When the VO reads << the other bit>> highlight red boxes. | |
| V3\_6 | Here, you will observe that the two bits of register ‘A’ with corresponding zero’s in register ‘B’ remain unchanged. | |  | | Display the image and when the VO reads <<the two bits>> highlight red box and then highlight green box. | |
| V3\_7 | From this example, you will notice that the bits of ‘A’ after the operation are obtained from the logic-OR operation of bits in ‘B’ and previous values of ‘A’. | | A 🡨 A V B 🡪 Logic-OR | | Display the image and type the TOS when the VO reads << logic-OR>>. | |
| V3\_8 | This example serves as a truth table from which you can deduce that selective-set operation is a logic-OR operation. | | |  |  |  | | --- | --- | --- | | **A** | **B** | **Output** | | 1 | 1 | 1 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 0 | 0 | 0 |   **Selective-set Operation** is a **Logic-OR Operation.** | | When the VO says << truth table >> Show the table and display the TOS in note strip. | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Screen No | | | | Heading / Title | | | | Video/Animation/Image No | | | | Integration -Description |
| 4 | | | | Selective-complement Operation | | | | Video 4 | | | | Video Template |
| Bulletin Label | | | | | | TOS | | | | | GD-Description | |
|  | | | | | |
| V4\_1 | | Let’s discuss the selective-complement operation performed in registers ‘A’ and ‘B’. | | | | Register A 🡪 1010  Register B 🡪 1100 | | | | | Animate the TOS as getting typed as per VO. | |
| V4\_2 | | In this operation, the bits in register ‘A’ are complemented where there are corresponding one’s in register ‘B’. It will not affect the bit positions that have zeros in register ‘B’. | | | | Bits in register A are complemented where there are corresponding 1’s in register B. | | | | | Show the TOS in note strip. | |
| V4\_3 | | In our example, the two leftmost bits of register ‘B’ are ones, so the corresponding bits of ‘A’ are complemented. | | | |  | | | | | Show the image and when the VO reads <<the two left>> highlight 1’s in red box.  When the VO reads <<corresponding>> highlight the blue. | |
| V4\_4 | | Here, you will observe that the two bits of register ‘A’ with corresponding zero’s in register ‘B’ remain unchanged. | | | |  | | | | | Display the image and when the VO reads <<the two bits>> highlight blue box and then highlight red box. | |
| V4\_5 | | From this example, you will notice that the bits of ‘A’ after the operation are obtained from the exclusive-OR operation of bits in ‘B’ and previous values of ‘A’. | | | |  | | | | | Display the images one by one. And highlight the red box. | |
| V4\_6 | | This example serves as a truth table from which you can deduce that selective-complement operation is an exclusive-OR operation. | | | | |  |  |  | | --- | --- | --- | | A | B | Output | | 1 | 1 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 0 | 0 | 0 |   **Selective-complement Operation** is an **Exclusive-OR Operation**. | | | | | When the VO says << truth table >> Show the table and display the TOS in note strip. | |
| Screen No | | | Heading / Title | | | | Video/Animation/Image No | | | Integration -Description | | |
| 5 | | | Selective-clear Operation | | | | Video5 | | | Video Template | | |
| Bulletin Label | | | | | TOS | | | | GD-Description | | | |
|  | | | | |
| V5\_1 | Let’s discuss the selective-clear operation performed in registers ‘A’ and ‘B’. | | | | Register A 🡪 1010  Register B 🡪 1100  Bits in register A are cleared where there are corresponding 1’s in register B. | | | | Animate the TOS as getting typed as per VO. | | | |
| V5\_2 | In this operation, the bits in register ‘A’ are cleared where there are corresponding one’s in register ‘B’.  It will not affect the bit positions that have zeros in register ‘B’. | | | | Show the TOS in note strip. | | | |
| V5\_3 | In our example, the two left most bits of register ‘B’ are ones, so the corresponding bits of ‘A’ are cleared. | | | |  | | | | Display the image and when the VO reads << the two>> highlight blue and then highlight red. | | | |
| V5\_4 | The other two bits of register ‘A’ with corresponding zero’s in register ‘B’ remain unchanged. | | | |  | | | | Display the image and when the VO reads <<The other two bits>> highlight blue box and then highlight red box. | | | |
| V5\_5 | From this example, we notice that the selective-clear operation is the logic-AND operation performed between ‘A’ and B-dash. | | | | |  |  |  | | --- | --- | --- | | A |  | Output | | 1 | 1 | 0 | | 0 | 1 | 0 | | 1 | 0 | 1 | | 0 | 0 | 0 |   **Selective-clear Operation** is the **Logic-AND Operation** performed between A and . | | | | Show the table and then show the TOS in note strip. | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Screen No | | Heading / Title | | Video/Animation/Image No | | Integration -Description |
| 6 | | Mask Operation | | Video6 | | Video Template |
| Bulletin Label | | | TOS | | GD-Description | |
|  | | |
| V6\_1 | Now, let us discuss the mask operation performed in registers ‘A’ and ‘B’. | | Register A 🡪 1010  Register B 🡪 1100  Bits in register A are cleared where there are corresponding 0’s in register B. | | Animate the TOS as getting typed as per VO. | |
| V6\_2 | The mask operation is similar to the selective-clear operation, except that the bits in register ‘A’ are cleared where there are corresponding zero’s in register ‘B’. | | Show the TOS in note strip. | |
| V6\_3 | In our example, the two rightmost bits of register ‘B’ are zeros, so the corresponding bits of ‘A’ are cleared. | |  | | Display the image and when the VO reads << the two>> highlight blue and then highlight red. | |
| V6\_4 | The two leftmost bits remain unchanged as the corresponding bits in ‘B’ are ones. | |  | | Display the image and when the VO reads <<The two left bits>> highlight blue box and then highlight red box. | |
| V6\_5 | From this example, you observe that the mask operation is like an AND micro-operation. | | |  |  |  | | --- | --- | --- | | A | B | Output | | 1 | 1 | 1 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 0 | 0 | 0 |   **Mask Operation** is an **AND Micro-Operation.** | | Show the table and show the TOS in note strip. | |
| V6\_6 | Most computers use mask operation than selective-clear operation. | | Most computers use mask operation than selective-clear operation. | | Show the TOS in note strip. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Screen No | | Heading / Title | | Video/Animation/Image No | | Integration -Description |
| 7 | | Insert Operation | | Video7 | | Video Template |
| Bulletin Label | | | TOS | | GD-Description | |
|  | | |
| V7\_1 | Let’s discuss the insert operation performed in eight bit registers ‘A’ and ‘B’ with values as shown on the screen. | | A 🡪 0110 1010  B 🡪 0000 1111  Masks the bits & then performs OR operation on them with the required value. | | Animate the TOS as getting typed as per VO. | |
| V7\_2 | The insert operation is done by masking the bits and then performing OR operation on them with the required value. | | Show the TOS in note strip. | |
| V7\_3 | When you replace the four leftmost bits of register ‘B’ by one-zero-zero-one, first we have to mask the unwanted bits. | | 1001 | | When the VO reads << register B>> show the TOS in yellow.  Then show the image and highlight the boxes in blue and red one by one. | |
| V7\_4 | Then, you have to insert the values and perform OR operation. | |  | | Show the images, when the VO reads << insert>> highlight red box and then highlight yellow box. | |
| V7\_5 | Here, you will observe that the insert operation is a sequence of AND and OR micro-operations. | | **Insert Operation** is a sequence of **AND & OR Micro-Operations.** | | Show the TOS in note strip. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Screen No | | Heading / Title | | Video/Animation/Image No | | Integration -Description |
| 8 | | Clear Operation | | Video8 | | Video Template |
| Bulletin Label | | | TOS | | GD-Description | |
|  | | |
| V8\_1 | Now, let us discuss about the clear operation performed in registers ‘A’ and ‘B’. | | Register A 🡪 1010  Register B 🡪 1010 | | Animate the TOS as getting typed as per VO. | |
| V8\_2 | The clear operation compares the bits in both the registers and produces an all zeros result if the two numbers are equal. | | Compares the bits in both the registers and produces an all 0 result if the two numbers are equal. | | Show the TOS in note strip. | |
| V8\_3 | In our example, the two corresponding bits are either both zero or one, and so the result will be all zeros. | |  | | Show the image and highlight red boxes from left to right and when the VO reads << the result>> highlight yellow box. | |
| V8\_4 | From this example, you observe that the clear operation is an exclusive-OR micro-operation. | | |  |  |  | | --- | --- | --- | | **A** | **B** | **Output** | | 1 | 1 | 0 | | 0 | 0 | 0 | | 1 | 1 | 0 | | 0 | 0 | 0 |   **Clear Operation** is an **Exclusive-OR Micro-Operation.** | | Show the table and show the TOS in note strip. | |
| V8\_5 | Thus, in all the examples discussed here, register ‘A’ is considered as a processor register and the bits of register ‘B’ constitute a logic operand extracted from the memory and placed in register ‘B’. | | Register A 🡪 Processor Register  Register B 🡪 Logic operand extracted from memory and placed in register B | | Show the TOS. | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Summary | | Screen No | Heading / Title | | Video/Animation/Image No | | Integration -Description |
|  | | 9 | Summary | | Video 9 | | Video template |
| Bulletin Label | | | | TOS | | GD-Description | |
|  | | | |
| V9\_1 | This brings us to the end of the lesson.  In this lesson, you have learnt that:  Logic micro-operations are used to perform bit style operations or manipulations on non-numeric data. | | | * Logic micro-operations are used to perform bit style operations or manipulations on non-numeric data | | Split the screen and show the bulletin points in info graphic style in the left side of the screen and highlight each point as per VO and show the corresponding image on the right side of the screen. | |
| V9\_2 | The selective-set operation is a logic-OR operation | | | * Selective-set operation is a logic-OR operation | |
| V9\_3 | The selective-complement operation is an exclusive-OR operation. | | | * Selective-complement operation is an exclusive-OR operation | |
| V9\_4 | The selective-clear operation is the logic-AND operation performed between one register and the complement of the other register. | | | * Selective-clear operation is the logic-AND operation performed between one register and the complement of the other register | |
| V9\_5 | The mask operation is an AND micro-operation. | | | * Mask operation is an AND micro-operation | |
| V9\_6 | The insert operation is a sequence of AND and OR micro-operations. | | | * Insert operation is a sequence of AND & OR micro-operations | |
| V9\_7 | The clear operation is an exclusive-OR micro-operation. | | | * Clear operation is an exclusive-OR micro-operation | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Evaluation | | Screen No | Heading / Title | | Video/Animation/Image No | | Integration -Description |
|  | | 10 | Evaluation | |  | | Evaluation Template |
| Bulletin Label | | | | TOS | | GD-Description | |
| V10\_1 | Evaluation | | | Evaluation | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Evaluation Content | | Screen No | | Heading / Title | Video/Animation/Image No | | Integration -Description |
|  | | 11 | | Drag and Drop |  | | Drag and Drop Template |
| Bulletin Label | | | TOS | | | GD-Description | |
| V11\_1 | Drag and drop the correct answer. | | |  |  |  | | --- | --- | --- | | S. No. | Questions | Answer | | 1. | The selective-complement operation is a/an \_\_\_\_\_\_\_\_\_ operation. | exclusive-OR | | 2. | The insert operation is done by \_\_\_\_\_\_\_ the bits and then performing \_\_\_\_\_ operation on them with the required value. | masking    OR | | 3. | Most computers use mask operation than \_\_\_\_\_\_\_\_\_\_ operation. | selective-clear | | | |  | |

Reference Details

|  |  |  |
| --- | --- | --- |
| Book/Website/Others | Author Name (Book) / URL (web reference) | Edition & Published by |
| Computer System Architecture | M.Morris Mano | Third edition |
|  | <http://www.niecdelhi.ac.in/uploads/Notes/btech/5sem/cse/CA_Notes.pdf> |  |
| *Important Note: It is mandatory that University referred books are used for content reference.* | | |

**Appendix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Language Verified by**: | | Sreedevi Rajagopal | | |
| VO Number | Error Type | | Corrections | Suggestions/Id Accepted or Not accepted the changes -Justify |
| V2\_1 | Intro to be reframed | | Comment Provided | ID not Accepted(this is a continuation of the previous topic, so I have given intro about the previous topic) |
| V3\_3 | Propositional Error | | Replaced | ID accepted |
| V5\_2 | Propositional Error | | Replaced | ID accepted |
| V6\_2 | Propositional Error | | Replaced | ID accepted |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |

|  |  |
| --- | --- |
| **Internal SME Review** | |
| SME Rating or Overall Review on SB | The content is good. |

|  |  |  |  |
| --- | --- | --- | --- |
| VO Number | Review Comments | Defect/Suggestion | ID accepted /ID not accepted – Justify |
| V\_title | Organization. | defect | ID not accepted(followed UK English) |
| V2\_3 | Use small letters itself as shown in the table. | suggestion | ID not accepted( captalised for VO) |
| V3 | Better to list out the applications that we are going to discuss. Then explanation one after the other. | suggestion | ID accepted |
| V3\_7 | If possible draw the truth table; and show whether the same logic applied or not.  A B output  0 0 0  0 1 1  1 0 1  1 1 1  Here A is set to 1 if the corresponding bit of B is 1.  Hence proved | suggestion | ID accepted |
| V4\_6 | Show the truth table or diagram for giving clear idea. As similar with previous comment, and for the remaining application also. | suggestion | ID accepted |
| V5\_1 | Selective-clear | defect | ID accepted |
| V7\_5 | Here the mask operation is an AND Operation and the insert operation is an OR micro operation.  Write like this.  It is sequence of AND then OR operation. It does not mean that it is both. | suggestion | ID accepted |
| V9\_6 | Previous comment(V7\_5) | suggestion | ID accepted |

|  |  |
| --- | --- |
| **External SME Review** | |
| SME - Overall Review on SB content | The content is good, do the changes mentioned. |
| SME Rating on SB quality | Good. |

|  |  |  |  |
| --- | --- | --- | --- |
| VO Number | Review Comments | Defect/Suggestion | ID accepted /ID not accepted – Justify |
| V\_title | Organization. | defect | ID not accepted(followed UK English) |
| V2\_3 | Use small letters itself as shown in the table. | suggestion | ID not accepted( capitalised for VO) |
| V3 | Better to list out the applications that we are going to discuss. Then explanation one after the other. | suggestion | ID accepted |
| V3\_7 | If possible draw the truth table; and show whether the same logic applied or not.  A B output  0 0 0  0 1 1  1 0 1  1 1 1  Here A is set to 1 if the corresponding bit of B is 1.  Hence proved | suggestion | ID accepted |
| V4\_6 | Show the truth table or diagram for giving clear idea. As similar with previous comment, and for the remaining application also. | suggestion | ID accepted |
| V5\_1 | Selective-clear | defect | ID accepted |
| V7\_5 | Here the mask operation is an AND Operation and the insert operation is an OR micro operation.  Write like this.  It is sequence of AND then OR operation. It does not mean that it is both. | suggestion | ID accepted |
| V9\_6 | Previous comment(V7\_5) | suggestion | ID accepted |

|  |  |
| --- | --- |
| **External SME QC Review – Pre SBQC** | |
| SME Rating or Overall Review on SB | The content is good. |

|  |  |  |  |
| --- | --- | --- | --- |
| VO Number | Review Comments | Defect/Suggestion | ID accepted /ID not accepted – Justify |
| V\_title | Organization. | defect | ID not accepted(followed UK English) |
| V2\_3 | Use small letters itself as shown in the table. | suggestion | ID not accepted(capitalised for VO) |
| V3 | Better to list out the applications that we are going to discuss. Then explanation one after the other. | suggestion | ID accepted |
| V3\_7 | If possible draw the truth table; and show whether the same logic applied or not.  A B output  0 0 0  0 1 1  1 0 1  1 1 1  Here A is set to 1 if the corresponding bit of B is 1.  Hence proved | suggestion | ID accepted |
| V4\_6 | Show the truth table or diagram for giving clear idea. As similar with previous comment, and for the remaining application also. | suggestion | ID accepted |
| V5\_1 | Selective-clear | defect | ID accepted |
| V7\_5 | Here the mask operation is an AND Operation and the insert operation is an OR micro operation.  Write like this.  It is sequence of AND then OR operation. It does not mean that it is both. | suggestion | ID accepted |
| V9\_6 | Previous comment(V7\_5) | suggestion | ID accepted |
|  |  |  |  |

Final QC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Review summary:**  *Evaluation scheme*: Definitely yes = A; mostly yes = B; mostly no = C; definitely no = D |  |  | |  |
| **Parameters** | **Internal Review Rating** | **External Review Rating** | | **Final QC** |
| **Relevance** *(Is the relevant content adequately explained addressing the syllabus, without making unreasonable assumptions about the students’ prior knowledge?)* | A | A | |  |
| **Creativity** *(Does the script address the topic in a non-conventional, imaginative and exciting way?)* | A | A | |  |
| **Sensitivity** *(Does the script try to gauge students’ existing conceptions of the topic/concept being addressed?)* | A | A | |  |
| **Pedagogy** *(Does the script encourage students to explore and discover?  Does it convey an image of the teacher as a facilitator rather than an authoritative figure?)* | A | A | |  |
| **Examples** *(Does it refer to real life contexts, use appropriate examples and/or analogies to help the students understand and relate to the topic)* | A | A | |  |
| **Questioning** *(Does the script has critical reasoning questions for the students to attempt? Are there formative and/or summative questions?)* | B | B | |  |
| **Animations** *(Does ample animations are there to aid learning – Minimum 3 )* | A | A | |  |
| **Applications** *(Does the Application(s)/Use/Need for learning this topic highlighted)* | A | A | |  |
| **Authentic and Adequate References** *( Have the ID referred the text books, reference books and Authentic websites like NPTEL, MIT OCW, Harvard, Code Academy, Courser)* | A | A | |  |
| **Syllabus and Examination Coverage** (Check the syllabus, Lesson Plan, Question bank, Question Paper and Books) | A | A | |  |
| **Total Number of Defects** | 2 | | 2 |  |
| **Total Number of Suggestions** | 6 | | 6 |  |
|  |  | |  |  |

|  |  |  |
| --- | --- | --- |
| **Final Qc Signed off by and on (Final QC SME name)** | Malla Sudhakar | 05/05/17 |

|  |  |
| --- | --- |
| **SME External QC Review – Final** | |
| Ex-QC Overall Review on SB content/video | Provide an overall feedback on the Storyboard Content/Video quality. |
| Ex-QC SME Rating | Excellent/Good/Average/Needs Improvement/Rework |

|  |  |  |  |
| --- | --- | --- | --- |
| VO Number | Review Comments | Defect/Suggestion | ID accepted /ID not accepted – Justify |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |