# **Logical Schema**

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Camping Car**

Camping Car – CAMP\_CAR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | CIN | VARCHAR (17) |  |  |  |  |
| 2 | ISSUE\_DT | DATE |  |  |  |  |
| 3 | CAR\_TP | VARCHAR (25) |  |  |  |  |
| 4 | MANU\_DT | DATE |  |  |  |  |
| 5 | DRI\_DIS | INTEGER |  |  |  |  |
| 6 | CAMP\_FAC | VARCHAR (50) |  |  |  |  |
| 7 | CAMP\_CAP | INTEGER |  |  |  |  |
| 8 | CAMP\_FL | TINYINT (1) |  |  |  |  |
| 9 | LCNS\_RQ | VARCHAR (10) |  |  |  |  |
| 10 | BIN | VARCHAR (18) |  |  |  |  |

specify the requirements for the pipelined microarchitecture. Second, we move on to concepts critically related to microarchitecture: multi-cycle microarchitecture, pipelined microarchitecture, performance analysis, data dependency handling, control dependency handling, and branch prediction. Third, we will state the data path, which includes the latches, forwarding unit, data hazard detection unit, branch prediction unit, and program definitions. Fourth, we will describe how we implemented the data paths and pipelined MIPS simulator according to the hardware components and program definitions. Then, there will be some results by executing the binary programs using an implemented simulator. In the end, we will evaluate the pipelined MIPS simulator with performance comparison with the single-cycle MISP simulator, and the flow of the data paths based on some assumptions.

# **Customer**

Customer – CUSTOMER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | LCNS\_NO | VARCHAR (12) |  |  |  |  |
| 2 | FNAME | VARCHAR (50) |  |  |  |  |
| 3 | MNAME | VARCHAR (50) |  |  |  |  |
| 4 | LNAME | VARCHAR (50) |  |  |  |  |
| 5 | CUS\_PHN | VARCHAR (11) |  |  |  |  |
| 6 | CUS\_EML | VARCHAR (50) |  |  |  |  |
| 7 | CUS\_ADDR | VARCHAR (50) |  |  |  |  |
| 8 | CUS\_AGE | INTEGER |  |  |  |  |
| 9 | LOGIN\_ID | VARCHAR (13) |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Customer Credential**

Customer Credential – CUSTOMER\_CREDENTIAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | LOGIN\_ID | VARCHAR (13) |  |  |  |  |
| 2 | PASSWORD | VARCHAR (13) |  |  |  |  |
| 3 | LOGIN\_TIME | TIME |  |  |  |  |
| 4 | LOGOUT\_TIME | TIME |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Event**

Event – EVENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | EIN | INTEGER AUTO INCREMENT |  |  |  |  |
| 2 | EVNT\_SDT | DATE |  |  |  |  |
| 3 | EVNT\_EDT | DATE |  |  |  |  |
| 4 | EVNT\_DES | VARCHAR (100) |  |  |  |  |
| 5 | APP\_LOC | VARCHAR (50) |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Accident**

Accident – ACCIDENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | AIN | VARCHAR (10) |  |  |  |  |
| 2 | CIN | VARCHAR (17) |  |  |  |  |
| 3 | ISSUE\_DT | DATE |  |  |  |  |
| 4 | ACC\_DT | DATE |  |  |  |  |
| 5 | ACC\_DES | VARCHAR (50) |  |  |  |  |
| 6 | DAMAGE | FLOAT |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Review**

Review – REVIEW

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | REV\_SQ | INTEGER AUTO INCREMENT |  |  |  |  |
| 2 | RATE | INTEGER |  |  |  |  |
| 3 | REV\_DES | VARCHAR (100) |  |  |  |  |
| 4 | REV\_CAR | VARCHAR (17) |  |  |  |  |
| 5 | LCNS\_NO | VARCHAR (12)S |  |  |  |  |
| 6 | CIN | VARCHAR (17) |  |  |  |  |
| 7 | ISSUE\_DT | DATE |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Accessory**

Accessory – ACCESSORY

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | ACSRY\_NO | VARCHAR (10) |  |  |  |  |
| 2 | STATUS | TINYINT (1) |  |  |  |  |
| 3 | ACSRY\_AMT | INTEGER |  |  |  |  |
| 4 | ACSRY\_TP | VARCHAR (50) |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Reservation**

Reservation – RESERVATION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | RID | VARCHAR (20) |  |  |  |  |
| 2 | RES\_SDT | DATE |  |  |  |  |
| 3 | RES\_EDT | DATE |  |  |  |  |
| 4 | RES\_LOC | VARCHAR (50) |  |  |  |  |
| 5 | PASS\_AMT | INTEGER |  |  |  |  |
| 6 | METR\_STD | INTEGER |  |  |  |  |
| 7 | METR\_END | INTEGER |  |  |  |  |
| 8 | RENT\_AMT | FLOAT |  |  |  |  |
| 9 | ADD\_AMT | FLOAT |  |  |  |  |
| 10 | TTL\_AMT | FLOAT |  |  |  |  |
| 11 | PNLT\_AMT | FLOAT |  |  |  |  |
| 12 | PAY\_TP | VARCHAR (10) |  |  |  |  |
| 13 | CRD\_NO | VARCHAR (16) |  |  |  |  |
| 14 | CRD\_NM | VARCHAR (50) |  |  |  |  |
| 15 | PAY\_AMT | FLOAT |  |  |  |  |
| 16 | BILL\_ADDR | VARCHAR (50) |  |  |  |  |
| 17 | LCNS\_NO | VARCHAR (12) |  |  |  |  |
| 18 | CIN | VARCHAR (17) |  |  |  |  |
| 19 | ISSUE\_DT | DATE |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Branch**

Branch – BRANCH

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | BIN | VARCHAR (18) |  |  |  |  |
| 2 | STATE | VARCHAR (20) |  |  |  |  |
| 3 | STREET | VARCHAR (20) |  |  |  |  |
| 4 | ZIP | VARCHAR (5) |  |  |  |  |
| 5 | BRN\_EML | VARCHAR (50) |  |  |  |  |
| 6 | BRN\_PHN | VARCHAR (11) |  |  |  |  |
| 7 | BRN\_CAP | INTEGER |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Insurance**

Insurance – INSURANCE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | IIN | VARCHAR (10) |  |  |  |  |
| 2 | INS\_TP | VARCHAR (15) |  |  |  |  |
| 3 | CLSN\_COV | TINYINT (1) |  |  |  |  |
| 4 | BODY\_COV | TINYINT (1) |  |  |  |  |
| 5 | MEDI\_COV | TINYINT (1) |  |  |  |  |
| 6 | INS\_PRC | FLOAT |  |  |  |  |
| 7 | CIN | VARCHAR (17) |  |  |  |  |
| 8 | ISSUE\_DT | DATE |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Choose**

Choose – CHOOSE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | RID | VARCHAR (20) |  |  |  |  |
| 2 | ACSRY\_NO | VARCHAR (10) |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.

# **Apply**

Apply – APPLY

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | RID | VARCHAR (20) |  |  |  |  |
| 2 | EIN | INTEGER |  |  |  |  |

The pipeline is the idea of dividing the instruction processing cycle into distinct processing stages. Also, it means that

microarchitecture processes a different instruction in each stage. Instructions consecutively in program order are processed in

consecutive stages.