# **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 | Variable | Type | Size | PK | FK | Index |
| 1 | Camping Car ID | CIN | VARCHAR | 17 | **☑** | **☑** |  |
| 2 | Issue date | ISSUE\_DT | DATE | 3 | **☑** | **☑** |  |
| 3 | Car type | CAR\_TP | VARCHAR | 25 |  |  |  |
| 4 | Manufacture date | MANU\_DT | DATE | 3 |  |  |  |
| 5 | Distance Driven | DRI\_DIS | INTEGER | 4 |  |  |  |
| 6 | Facilities | CAMP\_FAC | VARCHAR | 50 |  |  |  |
| 7 | Passenger | CAMP\_CAP | INTEGER | 4 |  |  |  |
| 8 | Availability | CAMP\_FL | TINYINT | 1 |  |  |  |
| 9 | Required License | LCNS\_RQ | VARCHAR | 10 |  |  |  |
| 10 | Branch ID | 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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | License number | LCNS\_NO | VARCHAR | 12 | **☑** | **☑** |  |
| 2 | First name | FNAME | VARCHAR | 50 |  |  |  |
| 3 | Middle name | MNAME | VARCHAR | 50 |  |  |  |
| 4 | Last name | LNAME | VARCHAR | 50 |  |  |  |
| 5 | Phone | CUS\_PHN | VARCHAR | 11 |  |  |  |
| 6 | Email | CUS\_EML | VARCHAR | 50 |  |  |  |
| 7 | Address | CUS\_ADDR | VARCHAR | 50 |  |  |  |
| 8 | Age | CUS\_AGE | INTEGER | 4 |  |  |  |
| 9 | Login ID | 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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Login ID | LOGIN\_ID | VARCHAR | 13 | **☑** | **☑** |  |
| 2 | Password | PASSWORD | VARCHAR | 13 |  |  |  |
| 3 | Login time | LOGIN\_TIME | TIME | 3 |  |  |  |
| 4 | Logout time | LOGOUT\_TIME | TIME | 3 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Event IDs | EIN | INTEGER AUTO INCREMENT |  | **☑** | **☑** |  |
| 2 | Start date | EVNT\_SDT | DATE | 3 |  |  |  |
| 3 | End date | EVNT\_EDT | DATE | 3 |  |  |  |
| 4 | Description | EVNT\_DES | VARCHAR | 100 |  |  |  |
| 5 | Damage | 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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Accident ID | AIN | VARCHAR | 10 | **☑** |  |  |
| 2 | Camping Car ID | CIN | VARCHAR | 17 | **☑** |  |  |
| 3 | Issue date | ISSUE\_DT | DATE | 3 | **☑** |  |  |
| 4 | Accident date | ACC\_DT | DATE | 3 |  |  |  |
| 5 | Description | ACC\_DES | VARCHAR | 50 |  |  |  |
| 6 | Damage | DAMAGE | FLOAT | 4 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Review ID | REV\_SQ | INTEGER AUTO INCREMENT |  | **☑** |  |  |
| 2 | Rate of the service | RATE | INTEGER | 4 |  |  |  |
| 3 | Description | REV\_DES | VARCHAR | 100 |  |  |  |
| 4 | Reviewed camping car | REV\_CAR | VARCHAR | 17 |  |  |  |
| 5 | License number | LCNS\_NO | VARCHAR | 12 |  |  |  |
| 6 | Camping Car ID | CIN | VARCHAR | 17 |  |  |  |
| 7 | Issue date | ISSUE\_DT | DATE | 3 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Accessory ID | ACSRY\_NO | VARCHAR | 10 | **☑** | **☑** |  |
| 2 | Status | STATUS | TINYINT | 1 |  |  |  |
| 3 | Amount | ACSRY\_AMT | INTEGER | 4 |  |  |  |
| 4 | Type | 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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** | **☑** |  |
| 2 | Start date | RES\_SDT | DATE | 3 |  |  |  |
| 3 | End date | RES\_EDT | DATE | 3 |  |  |  |
| 4 | Start location | RES\_SLOC | VARCHAR | 50 |  |  |  |
| 5 | End location | RES\_ELOC | VARCHAR | 50 |  |  |  |
| 6 | Passenger | PASS\_AMT | INTEGER | 4 |  |  |  |
| 7 | Meter start | METR\_STD | INTEGER | 4 |  |  |  |
| 8 | Meter end | METR\_END | INTEGER | 4 |  |  |  |
| 9 | Rental amount | RENT\_AMT | FLOAT | 4 |  |  |  |
| 10 | Additional amount | ADD\_AMT | FLOAT | 4 |  |  |  |
| 11 | Total amount | TTL\_AMT | FLOAT | 4 |  |  |  |
| 12 | Penalty amount | PNLT\_AMT | FLOAT | 4 |  |  |  |
| 13 | Payment type | PAY\_TP | VARCHAR | 10 |  |  |  |
| 14 | Card number | CRD\_NO | VARCHAR | 16 |  |  |  |
| 15 | Card name | CRD\_NM | VARCHAR | 50 |  |  |  |
| 16 | Pay amount | PAY\_AMT | FLOAT | 4 |  |  |  |
| 17 | Billing address | BILL\_ADDR | VARCHAR | 50 |  |  |  |
| 18 | License Number | LCNS\_NO | VARCHAR | 12 |  |  |  |
| 19 | Camping Car ID | CIN | VARCHAR | 17 |  |  |  |
| 20 | Issue date | ISSUE\_DT | DATE | 3 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Branch ID | BIN | VARCHAR | 18 | **☑** | **☑** |  |
| 2 | State | STATE | VARCHAR | 20 |  |  |  |
| 3 | Street | STREET | VARCHAR | 20 |  |  |  |
| 4 | Zip | ZIP | VARCHAR | 5 |  |  |  |
| 5 | Email | BRN\_EML | VARCHAR | 50 |  |  |  |
| 6 | Phone | BRN\_PHN | VARCHAR | 11 |  |  |  |
| 7 | Capacity | BRN\_CAP | INTEGER | 4 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Insurance ID | IIN | VARCHAR | 10 | **☑** |  |  |
| 2 | Type | INS\_TP | VARCHAR | 15 |  |  |  |
| 3 | Collision coverage | CLSN\_COV | TINYINT | 1 |  |  |  |
| 4 | Body coverage | BODY\_COV | TINYINT | 1 |  |  |  |
| 5 | Medical coverage | MEDI\_COV | TINYINT | 1 |  |  |  |
| 6 | Price | INS\_PRC | FLOAT | 4 |  |  |  |
| 7 | Camping car ID | CIN | VARCHAR | 17 |  |  |  |
| 8 | Issue date | ISSUE\_DT | DATE | 3 |  |  |  |

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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** |  |  |
| 2 | Accessory ID | 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 | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** |  |  |
| 2 | Event IDs | EIN | INTEGER | 4 |  |  |  |

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.

1. **PAY\_TYPE**

Pay type –PAY\_TYPE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Payment type | PAY\_TP | INTEGER AUTO INCREMENT |  | **☑** | **☑** |  |
| 2 | Payment type name | PAY\_NM | VARCHAR | 10 |  |  |  |

1. **PAY**

Pay –PAY

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) | Index |
| 1 | Payment ID | PAY\_ID | VARCHAR | 10 | **☑** | **☑** |  |
| 2 | Payment type | PAY\_TP | VARCHAR | 10 |  |  |  |
| 3 | Payment amount | PAY\_AMT | FLOAT | 4 |  |  |  |
| 4 | Payment date | PAY\_DT | DATE | 3 |  |  |  |