Data Transfer Scheme :-

① (page number)

Microprocessor system में various speeds की अनेक I/o devices हो सकते हैं। Slow speed device उस समय Data transfer नहीं कर सकती जब Microprocessor इसके लिए instruction देता है क्योंकि device को ready होने में कुछ समय लगता है। Microprocessor एवं I/o device की speed के इस difference या Mismatch के कारण Data transfer की अनेक विधियाँ विकसित की गई हैं। इन्हें सामान्य रूप से दो वर्गों में divide किया जा सकता है :-

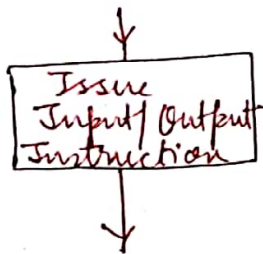① Programmed Data Transfer ② Direct memory Access Data Transfer (DMA)

Programmed Data Transfer :-

Programmed Data Transfer विधि CPU द्वारा Control होती है I/o device से CPU को अथवा CPU से I/o device को Data transfer, Microprocessor की Memory (ROM) में पहले से load किए गए Program द्वारा control होता है। CPU इन Program को उस समय Execute करती है जब I/o device data transfer करने के लिए ready हो जाती है। Program data transfer विधि का उपयोग तब किया जाता है जब कम मात्रा में Data transfer करने की आवश्यकता होती है।
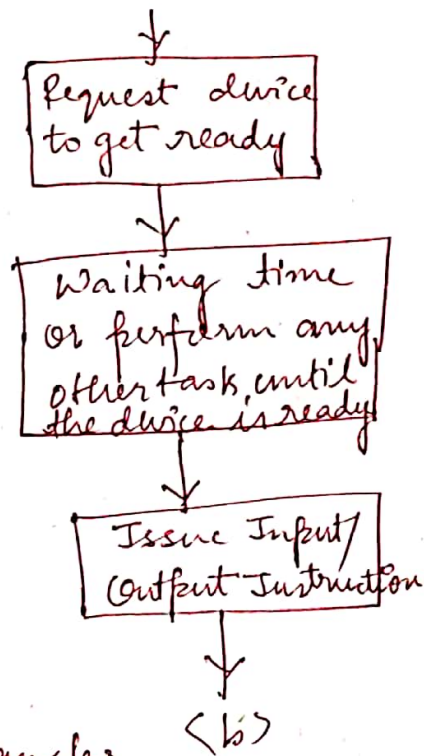
Program Data Transfer को तीन प्रकार Classify किया जाता है :-

① Synchronous Data Transfer ② Asynchronous Data Transfer

③ Interrupt driven Data Transfer

① <u>Synchronous Data Transfer</u> :- Data Transfer की यह सबसे सरल योजना है। यह विधि तब प्रयोग की जा सकती है जब I/o युक्ति तथा Microprocessor की गति परस्पर Match करती है। Device से Data Transfer करने के लिए Device को उचित instruction दी जाती है। Generally, इस प्रकार की I/o युक्तियाँ उपलब्ध नहीं होती जो कि Microprocessor से Completely compatible हो, परन्तु यदि I/o device की speed, Microprocessor से Match न करती हो तब भी Synchronous Data transfer method का उपयोग किया जा सकता है।

```
┌──────────────┐
│ Issue        │
│ Input/Output │
│ Instruction  │
└──────────────┘
```

〈a〉

```
┌──────────────┐
│ Request device│
│ to get ready  │
└──────────────┘
        │
┌──────────────┐
│ Waiting time │
│ or perform any│
│ other task, until│
│ the device is ready│
└──────────────┘
        │
┌──────────────┐
│ Issue Input/ │
│ Output Instruction│
└──────────────┘
```
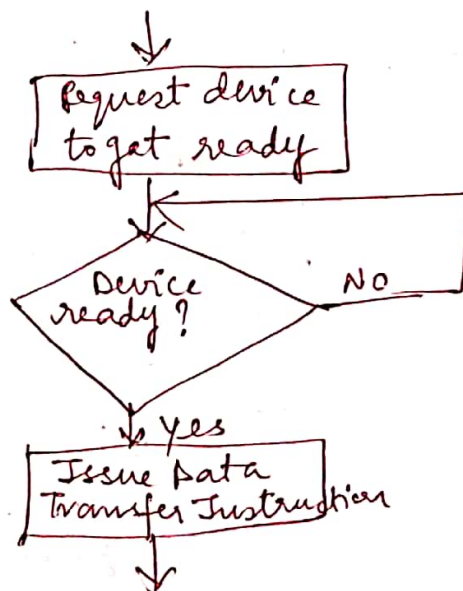
〈b〉

Synchronous Data transfer

इस प्रकार Synchronous Data transfer दो प्रकार से किया जा सकता है तथा यह Device की speed पर depend करता है। Above fig में दो विधियाँ दिखाई गई हैं। चित्र 〈a〉 में प्रदर्शित विधि में Device की speed match होना आवश्यक है जो कि मुश्किल से होता है। चित्र〈b〉 में Device की speed mismatch है अतः पहले request signal किया जाता है। Synchronous Data Transfer Method को अकेले प्रायः प्रयोग नहीं किया जाता। यह, generally अन्य विधियों के साथ प्रयुक्त किया जाता है लेकिन speed match होने पर Data transfer के लिए इसी विधि का उपयोग किया जाता है।

Asynchronous Data Transfer

```
┌──────────────┐
│ Request device│
│ to get ready  │
└──────────────┘
        │
     ◇ Device ready? ──NO──┐
        │ yes              │
┌──────────────┐           │
│ Issue Data   │───────────┘
│ Transfer Instruction│
└──────────────┘
```
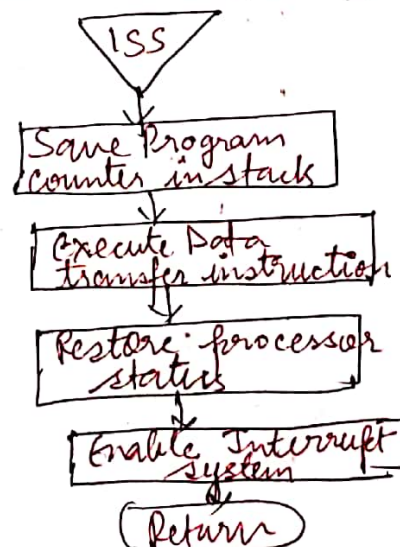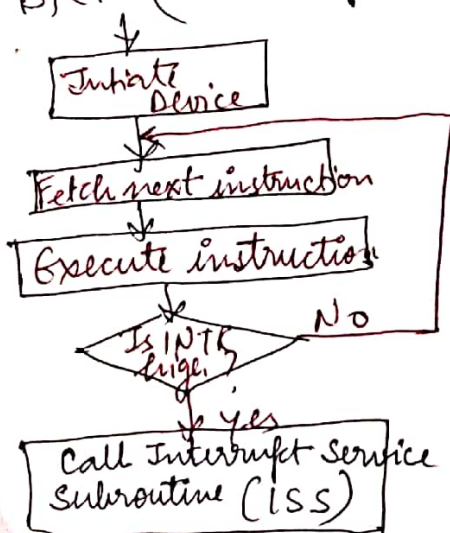
जब I/o device तथा microprocessor की speed match नहीं करती तब Asynchronous विधि से Data Transfer किया जाता है।

इस विधि में Data Transfer किसी पूर्व-निर्धारित time pattern के अनुसार नहीं होता। इस विधि में Data transfer करने से पहले, Microprocessor I/o device का status check करता है कि Device Data Transfer के लिए ready है अथवा नहीं। Micro-processor I/o device को पहले 'get ready' signal भेजता है तथा I/o device का status लगातार check करता रहता है जब तक कि I/o device Data Transfer के लिए ready नहीं होती। जब ready हो जाती है तब Microprocessor Data Transfer Instruction भेजता है।

Interrupt Driven Data Transfer

Interrupt method द्वारा Data transfer का प्रयोग तब किया जाता है जब I/o device की speed microprocessor की speed से कम होती है। इस method में Microprocessor सर्वप्रथम I/o device को initiation का संकेत देता है तथा अपने main program को execute करता है। इस अवधि में वह Program loop में आकर I/o device का status check नहीं करता। जब I/o device data transfer के लिए ready हो जाता है तब यह interrupt line (INTR) द्वारा microprocessor को high signal देती है। Interrupt signal receive करने के बाद micro-processor PC (Program Counter) के Content, stack में save करता है तथा उसके पश्चात एक subroutine पर जाता है। जिसे (Interrupt Service Subroutine or ISS) कहते हैं।

```
        │                          ▼
        ▼                        ┌───┐
┌──────────────┐                 │ISS│
│  Initiate    │                 └─┬─┘
│  Device      │                   ▼
└──────┬───────┘            ┌──────────────┐
       ▼                    │ Save Program │
┌──────────────────┐        │Counter in stack│
│Fetch next instruction│    └──────┬───────┘
└──────┬───────────┘               ▼
       ▼                    ┌──────────────┐
┌──────────────────┐        │ Execute Data │
│Execute instruction│       │transfer instruction│
└──────┬───────────┘        └──────┬───────┘
       ▼            No              ▼
    ◇───────────◇           ┌──────────────┐
   ◇ Is INTR    ◇──────     │Restore processor│
   ◇  high?     ◇           │ status       │
    ◇───────────◇           └──────┬───────┘
       │ Yes                        ▼
       ▼                    ┌──────────────┐
┌──────────────────┐        │Enable Interrupt│
│Call Interrupt Service│    │ System       │
│Subroutine (ISS)  │        └──────┬───────┘
└──────────────────┘               ▼
                              ( Return )
```

12/04/2020