Implementing Procedures

- Duplicate on demand:
  - high overhead for copying
  - large graphs
  - recursion implies the need for infinite graph storage.
- Provide a single graph and annotate the tokens entering it. This is called colouring.

Lecture 18

General purpose data-flow, procedures recursion and the Manchester architecture.

Colouring

- Change the operation firing rule to select an operation for excution when it has inputs of the same colour available on all its input arcs.
- Data items entering a procedure subgraph are coloured as they enter and de-coloured when they leave.

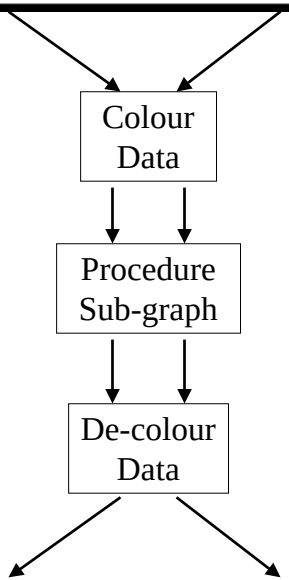
General purpose data-flow

- Hardware data-flow has a number of problems;
  - single use programs
  - no support for recursion/procedures
  - cannot multitask
- General purpose data-flow requires;
  - matching data to operations
  - representing the graph
  - scheduling operations when the data is available.

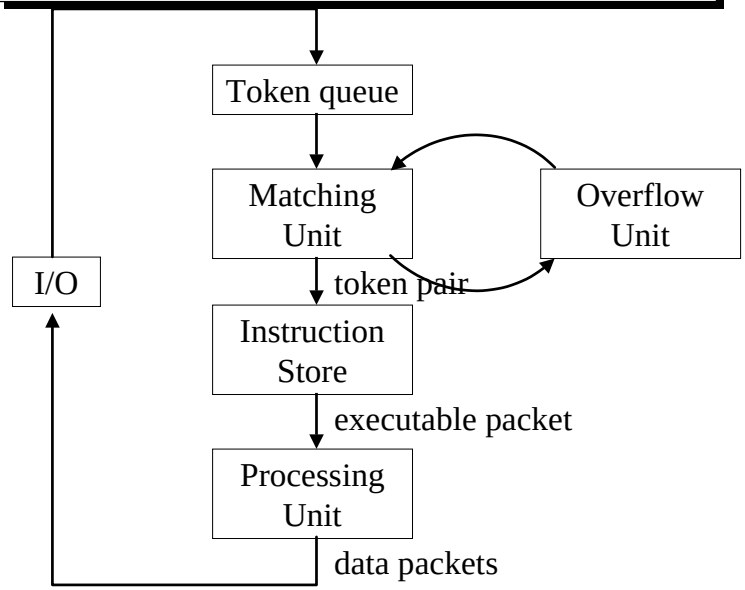
Manchester Architecture

- Assumptions:
  - binary and unary operators, to simplify data matching.
  - asynchronous execution on a cyclic architecture.
  - different elements of the ring can execute at different rates.
  - one bottleneck is the speed of the matching unit.

Colouring



Manchester Architecture



Manchester Architecture

- Attempts to provide general purpose support for data-flow programming.
  - processor pool (20 microcoded cpus)
  - associative matching of data
  - instruction store for operations.
- Executed 1-2 Million Instructions per second, and was running in 1985.

Matching Unit

- Receives a token from the buffer:
- Uses,
  - colour
  - value
  - destination node address
- Determines the matching partner token, if that token is stored in the matcher already.
- Uses associative memory to do matching.
- Overflow causes huge overheads.

I/O Switch

- Connects the system to a host computer from which programs and data are initially loaded, and to which the results are returned.

Instruction Store

- Decodes token destination operation address.
- Loads the relevant operator and appends it to the tokens to form an executable packet.

Token Queue

- Provides buffering of tokens
- Token generation is often bursty.
- Storage capacity 32K tokens.

Processing Unit

- Up to 20 microcoded cpus.
- Round robin scheduling of packets to processors.
- Cpus generate up to two output packets containing:
  - a value
  - a colour
  - a destination