READOUT OF A REMOTE IMAGING SPECTROMETER

Douglas A. Jensen
University of Massachusetts at Amherst
Amherst Mass. 01003

Abstract

A readout scheme for the remote imaging spectrometer is outlined. First a simplified system for a small spectrometer is developed in some detail. The changes necessary to upgrade this simple model to facilitate a larger area spectrometer and to facilitate the processing of multiple tracks per microbunch are discussed. The object of this exercise is to demonstrate the viability of remote imaging vis-a-vie the electronic readout and processing.

Introduction

It has been suggested[1] that some of the problems associated with the high fluxes necessary to study 'b' physics may be circumvented, or at least mitigated, if the vertex detector spectrometer is remote from the production target. This spectrometer is discussed in some detail in the articles by Bjorken[1] and by Wehman[2] in these proceedings.

The spectrometer would consist of a number of 'sub-spectrometers, (subsequently referred to as spectrometers) each in turn consisting of detector stations (nominally 3). The spectrometer would yield two pairs of points (X_1,Y_1; X_2,Y_2) at the front and rear of the spectrometer. These parameters are then used to calculate the Map Function (see article by A. Wehman[2]) Real-time limits imposed on this map function from the two spectrometer arms would then be used to select events with vertices away from the production target - the 'b' signature.

It is the object of this discussion to suggest a readout and processing scheme that will facilitate accepting and processing an event in each microbunch (every 18 nsec). For reasons of time and space, the system to be discussed here will be for a 'small' spectrometer which is able to process one hit per microbunch. The modifications necessary to build a large high resolution system and necessary to process multiple events per microbunch will be discussed briefly at various points in the discussion. In this discussion, 'spectrometer' will refer to the detector in one arm. Of course, two arms are necessary to detect the two particles from an interesting event.

Spectrometer

The spectrometer in each arm of the full two-body spectrometer consists of a number of overlapping spectrometers, each with a momentum acceptance of 10%. Each spectrometer consists of 3 readout stations. Each readout station consists of three (or 4) planes of Si detectors about 6 cm^2. These three detectors are: one (or preferably 2) rectangular Si pad arrays with coarse resolution; and an 'x' and a 'y'

array of high resolution Si strips. The high resolution strips would be at say 25 micron spacing. The rectangular pads would 'cover' between 32 and 128 of the fine strips. The layout of the detector and of a station is shown in figure 1 (for a small spectrometer, see below).

In the readout scheme outlined below, a track (or tracks) would be found both views using the pads. A maximum of one track passing through a particular set of pads would be accepted. In the multiple track scheme, it would be possible to accept multiple tracks only if the tracks occur in distinct sets of pads. (A particular pad may be involved in more than one 'pad-track': see the example tracks in fig 1.) This limitation is imposed to simplify the pattern recognition problem. With this limitation, there is no multiple track ambiguity problem within the one set of pads (an area of a few mm^2), i.e., there is no multiple track problem in the high resolution track finding. This limitations not considered serious.

Small Spectrometer

For purposes of discussion, consider a set of detectors with 8 x 8 pads (one set only) with each pad being 'backed up' by 8 high resolution strips. As shown in fig. 1, each station thus consists of 64 pads and 2 x 8 x 8 = 128 fine strips. The data is assumed to be in digital form on cables arriving from detector electronics at each station. Readout schemes for Si detectors are discussed elsewhere, as well as in these proceedings.

The first step in processing is to determine if there is a track in the pads. See Figure 2. This step will take a few clock cycles (1 clock = the microbunch time of 18 nsec). During this time, the data from the high resolution strips 'waits' in the cables.

The pads are assumed to be arrayed in rows (x) and columns (y). The first step in processing is form the 'or' of the 'y' pads to determine the presence of hits in 'x', and then to encode (using priority encoders) the 'x' addresses of the hits. This encoding is done simultaneously in the three stations. (The data from the pads must be sampled, and then be allowed to continue through cables to the 'y' encoding and track finding.) These 'x' pad hits are then tested to see if they form a straight line within the resolution of the pads. Note that it is often necessary to accept either of two 'x' pads as the confirming hit. This possibility is allowed for in the downloaded lookup table (or hard-wired look-up table). This facility is represented by the two comparators in figure 2. If no track is found in this view, the event is dropped. If a track is found in 'x', the 'y' pad arrays in which the track was found are encoded and searched for a track using exactly the same algorithm and hardware as described above. The differences in the 'x' and 'y' schemes are noted by the differences in the box around the input for station n+2 and the box to the right in figure 2. Again, if the track is not found in this view, the event is dropped. Note that as only the appropriate 'y' columns are searched, once a pad-track is found, it is fully correlated in x-y.

If a track has been found in both views, the clusters of high resolution strips associated with the row and column pads are encoded is added to obtain a position. These high resolution numbers are then used to determine the presence of a track. The logic is essentially the same as that for the pad track finding shown in figure 2. Note that once a track has been confirmed in each view, it is not really necessary to do a least squares fit, as the resolution of the track parameters would not be significantly improved. Note also that if the number of wires behind each pad is equal to the number of pads, the encoding hardware in the high resolution section is the same as in the pad section. This scheme can be continued in a full scale system, thus simplifying and expediting the design and construction. If the reconstruction of a high resolution track is successful in each view, the event is kept, and the event is passed on to be saved and further processed. Otherwise this spectrometer element reports nothing out.

In order to accept events at a rate of one per microburst, it is necessary to pipeline this scheme. Input data and results of calculations must be buffered at each clock time so that a new event may enter. This buffering and timing scheme is as follows.

At the first stage, when the columns are 'or'ed and encoded, and the encoded addresses of the pads are stored in registers. Even with a large system, it is relatively easy to carry out this step in 18 nsec, as the 'or' functions require about 1 nsec/gate, and the priority encoders with latches are quite fast. (MC10H165 for example require about 2.5 nsec. A 64 bit priority encoder scheme is presented in the Mororola 'Mecl Device Data' handbook). These row addresses must be propagated through the full system. During the next clock cycle, the lookup table is interrogated and the results buffered. At a third clock pulse, the results of the lookup table are compared with the confirming plane buffer and a decision is made to proceed or reject the event. Data is always passed forward into a new register to make room for the next event.

If the event is kept, at the fourth clock pulse, the appropriate columns 'y' columns are selected and encoded. (Again, an example of a 64 bit data selector, appropriate to a full sized system of 64 x 64 pads, is shown in the above mentioned Mororola handbook.) These encoded addresses are processed in the same system as above. (now up to 6 clock cycles) At this point, again assuming that the event is to be kept, the 'x' and 'y' pad addresses from the 3 stations that form a pad-track are available. The process of finding high resolution tracks can then proceed in parallel in the two views, as the x-y correlation is already assured by the pad-tracks.

Note that if multiple tracks are to be allowed in a spectrometer, considerable extra hardware must be added. At each successive stage of track searching, the full pad image must be available. The first set of pads used in a track must be turned off, and the encoding cycle started again. The 'y' search is even more complex, as at that point, events with possible ghost solutions would have to be rejected. Multiple track finding will not be further considered in this brief discussion.

Calculating the Map Function

Once the track finding is complete (values of X_n, Y_n; X_{n+2}, Y_{n+2}; generally noted here and in ref. 2 as X_1, Y_1; X_2, Y_2; are available), the track parameters are passed on to the Map Function logic. Depending upon the rates of the output of track finding, it may be appropriate to multiplex this part of the spectrometer, as it is probably going to be expensive! The function to be calculated is a sum of terms of the type:[2]

$C_i X_1^{p1} Y_1^{p2} X_2^{p3} Y_2^{p4}$; p1+p2 odd, p3+p4 odd where the order of the function is n where

$n = p1 + p2 + p3 + p4$. For n = 6, there are 59 terms! and of course 59 C_i.

The map function to sixth order consists of three sets of terms:, the second order (3 of the 10 possible second order terms enter), 19 fourth order, and 39 sixth order. It is therefore logical to pipeline the calculation in this order. At each stage of the pipeline calculation, a set of terms of the type $C_i F_a F_b$ must be calculated, where each of the F's at a particular stage are functions of two of the F's calculated at the previous stage (X_1 ... are the 0th stage). These functions are probably best done by lookup table, as it is then easy to generate more complex functions than simple multiplies. (At very least, constants must be included.) The contribution to the map would then be the sum of terms contributing to the map function from the previous stage(s), plus terms from the current stage. One stage of this calculation is outlined in Figure 3.

Each pipeline step might have 10 such multiply units. All possible second order terms would then be calculated in one unit. (Only 3 of the second order terms are needed in the map function, but others are needed in the higher order terms.) Two multiply units would be needed in parallel to do the fourth order calculation, and 4 would be used to calculate the sixth order terms. 11 input adders would be necessary to accumulate the sums as the calculation proceeds through the pipeline. The memories for these calculations, are in the most naive model, very large: about 2M words each or about 20M words per 10 multiplier unit!. Seven of these units would be required for the full system. The word size for each unit depends on the accuracy needed, and that of course sets the size of the next stage of memory. The estimation of 2M words per multiply unit assumes that the inputs to the units would be 11 bit words (approximately the number of address bits to specify a high resolution Si strip) There are of course a number of ways to reduce the amount of logic necessary. For example, partial products may be calculated and summed to form higher precision products (a technique used in BNL-E766, and to be used in the FNAL-E690 hardware processor). Some of the calculations may be eliminated based on a detailed analysis of the terms in the map.

In order to monitor, calibrate, and download constants, it is of course necessary to have this system controlled by and monitored by a computer. This connection is not shown in figures 2 and 3. It is assumed that a number of events can be passed to this on-line computer in real time for continuous real-time monitoring and calibration. Adjusted tables of constants can be reloaded during the time between beam spills, or at appropriate intervals.

Also not included in this discussion is the hardware to correlate the tracks found in the two arms of the spectrometer. This correlation would be carried out in another intelligent device using the outputs of the map calculator from each of the spectrometer arms.

Conclusions

It is possible to construct a system to accept and process an event per microbunch. The track finding is quite straightforward. The map calculations, if indeed of order 60 terms are needed, and if the terms must be completely general, is an expensive project in terms of the amount of hardware. This expense may be mitigated by multiplexing to reduce the number of map calculators per spectrometer arm, and possibly by judicious formulation of the map calculation.

I am indebted to Bjorken and Wehman for detailed discussions of this proposal. This work supported by NSF grant number 11052 (University of Mass.)

1) " A Minimal Experiment" by Bjorken, Transparencies available at this workshop. See also the article in these proceedings.

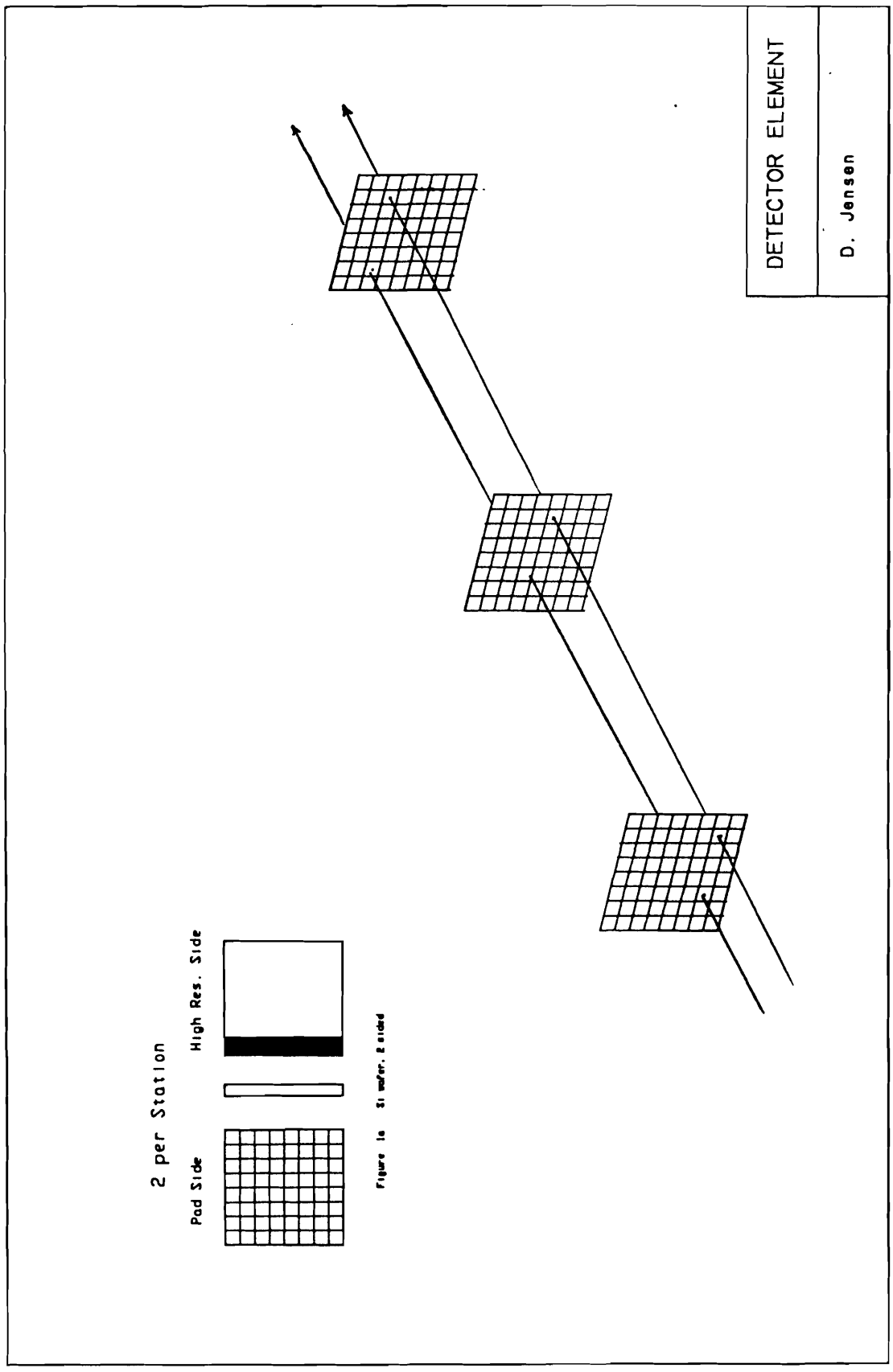
DETECTOR ELEMENT
D. Jensen
2 per Station
Pad Side
High Res. Side

Figure 1

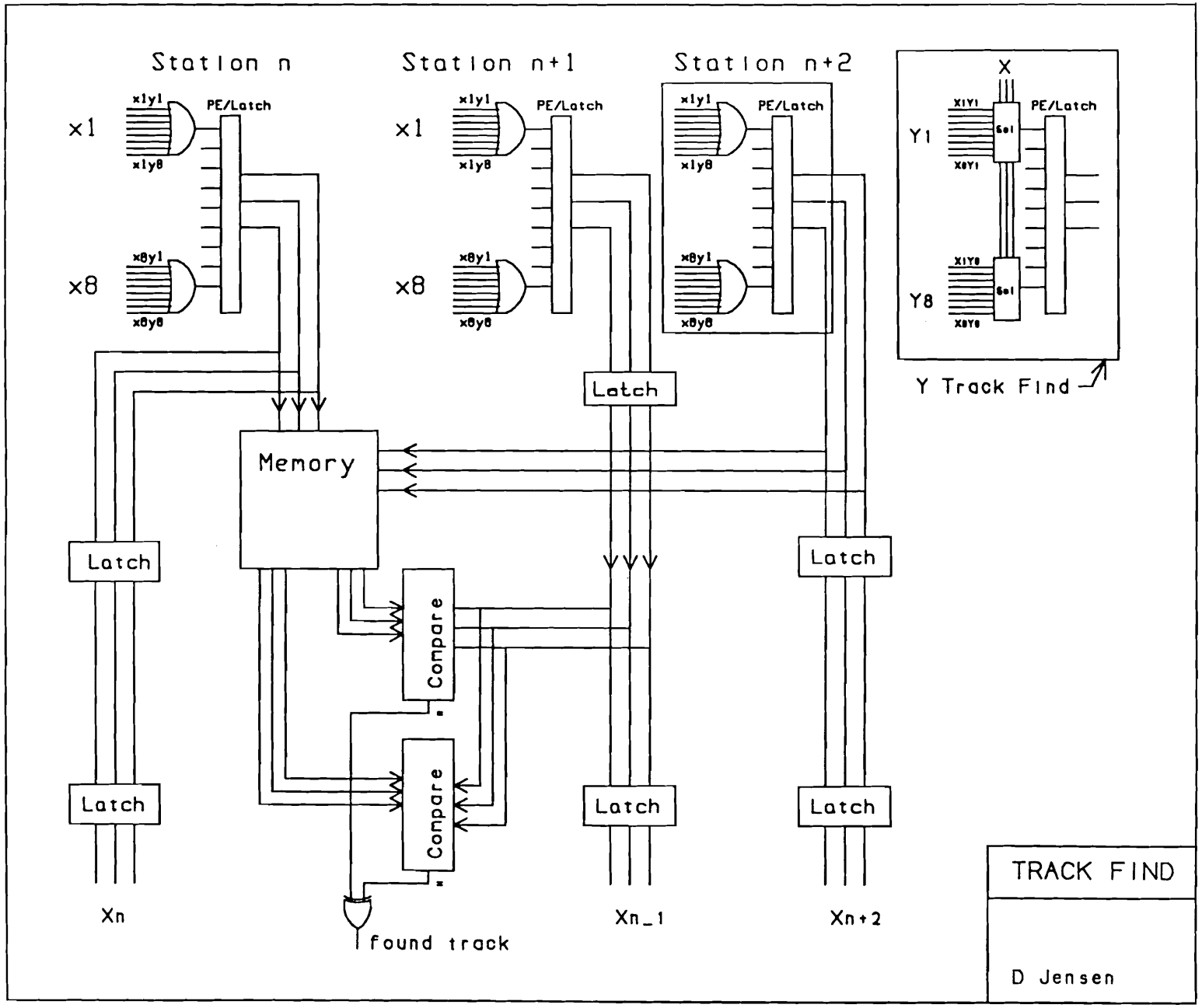
Station n
Station n+1
Station n+2
x1
x8
x1y1
x1y8
x8y1
x8y8
PE/Latch
X
Y1
Y8
X1Y1
X8Y1
X1Y8
X8Y8
Sel
Y Track Find
Memory
Latch
Compare
Xn
Xn_1
Xn+2
found track
TRACK FIND
D Jensen

Figure 2

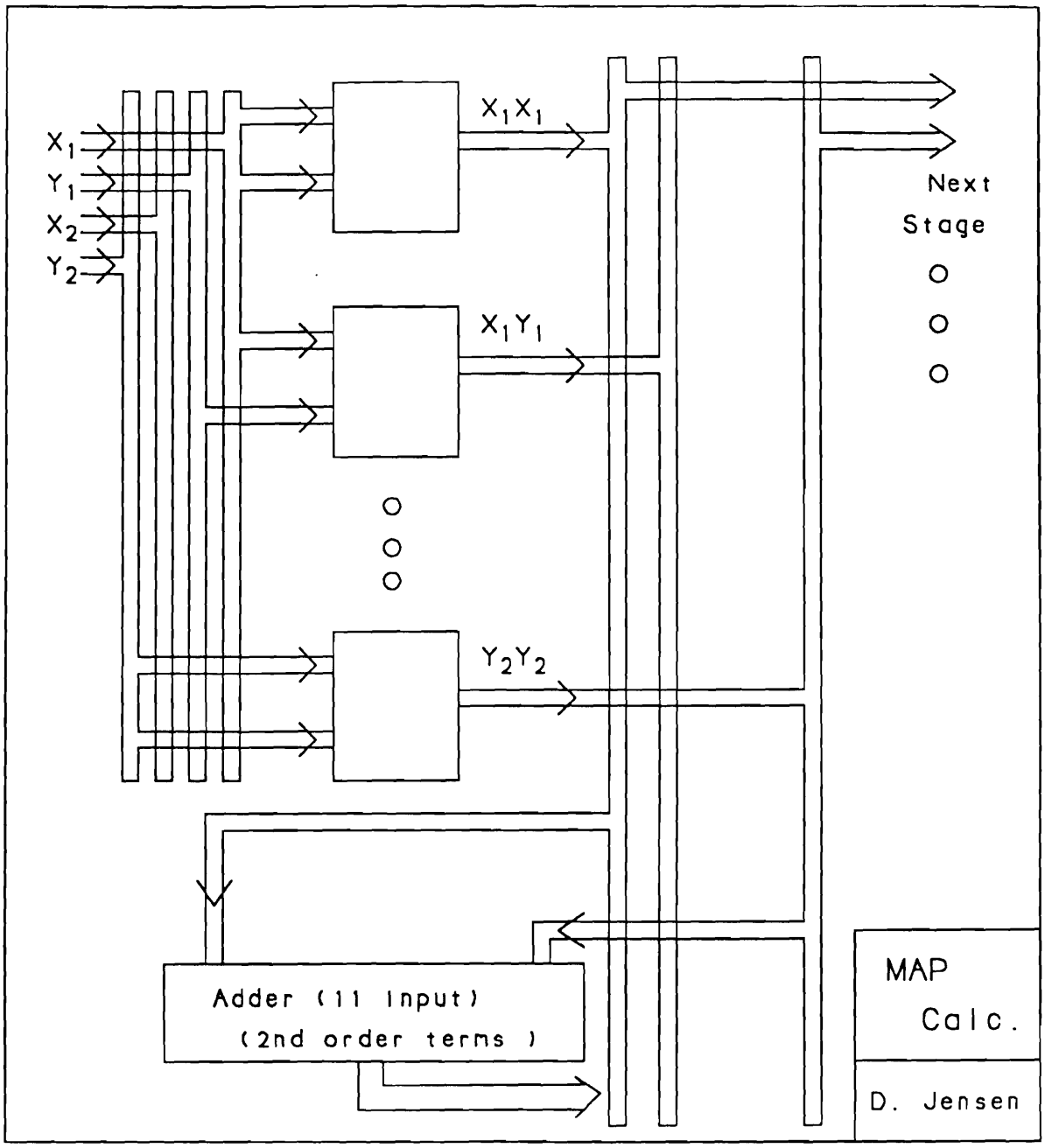
X1
Y1
X2
Y2
X1X1
X1Y1
Y2Y2
Next
Stage
Adder (11 input)
(2nd order terms)
MAP
Calc.
D. Jensen

Figure 3